

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

# **Návrh a realizace Wi-Fi datového komunikátoru pro IQRF síť s ohledem na spotřebu elektrické energie**

## **Design and Implementation of Wi-Fi Data Communicator for IQRF Networks with Respect to Energy Consumption**

## Zadání diplomové práce

Student:

**Bc. Miroslav Mikuš**

Studijní program:

N2649 Elektrotechnika

Studijní obor:

2612T041 Řídicí a informační systémy

Téma:

Návrh a realizace Wi-Fi datového komunikátoru pro IQRF síť s ohledem  
na spotřebu elektrické energie  
Design and Implementation of Wi-Fi Data Communicator for IQRF  
Networks with Respect to Energy Consumption

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Popis technologie IQRF a mikrokontroléru NXP Kinetis řady L.
2. Popis možností přenosu dat ze sítě IQRF na datové uložení.
3. Návrh a realizace hardware IQRF/Wi-Fi komunikátoru s využitím mikrokontroléru NXP Kinetis řady L.
4. Návrh a realizace firmware pro daný komunikátor.
5. Zprovoznění komunikace mezi IQRF/Wi-Fi komunikátorem a datovým uložením.
6. Vytvoření komunikačního protokolu pro nastavování vlastností komunikátoru.
7. Testování vytvořeného zařízení s ohledem na spotřebu elektrické energie.
8. Zhodnocení výsledků a závěr.

Seznam doporučené odborné literatury:

- [1] JACKO, Róbert. *Energeticky soběstačný bezdrátový modul pro senzorické aplikace*. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. Dostupné také z: <http://hdl.handle.net/11012/32913>.
- [2] BRTNÍK, Bohumil a David MATOUŠEK. *Algoritmy číslicového zpracování signálů*. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-400-2.
- [3] DOSTÁLEK, Libor a Alena KABELOVÁ. *Velký průvodce protokoly TCP/IP a systémem DNS*. 2. vyd. Computer Press Praha, 2000. 426 s. ISBN 80-7226-323-4.
- [4] NXP Semiconductors. *KL28Z Reference Manual*. 2016. [online] Dostupné z: <http://www.nxp.com>.
- [5] MICRORISC s.r.o. *SPI Implementation in IQRF For (DC)TR-7xD*. 2016. [online] Dostupné z: <http://www.iqrf.org>.
- [6] MICRORISC s.r.o. *IQRF Quick Start Guide For IQRF OS v3.07D and higher*. 2017. [online] Dostupné z: <http://www.iqrf.org>.
- [7] MICRORISC s.r.o. *IQRF OS Operating System Version 4.00D for (DC)TR-7xD - User's Guide*. 2017. [online] Dostupné z: <http://www.iqrf.org>.
- [8] MICRORISC s.r.o. *IQRF DPA Framework Technical Guide v3.00*. 2017. [online] Dostupné z: <http://www.iqrf.org>.
- [9] MICRORISC s.r.o. *IQRF Cloud Technical guide*. 2015. [online] Dostupné z: <http://www.iqrf.org>.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Martin Pieš, Ph.D.**

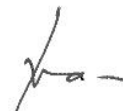
Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2019



---

doc. Ing. Jiří Koziorek, Ph.D.  
*vedoucí katedry*




---

prof. Ing. Pavel Brandštetter, CSc.  
*děkan fakulty*

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární  
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2019

..........



Na tomto místě bych rád poděkoval zejména svému vedoucímu Ing. Martiu Piešovi, Ph.D za příkladné pedagogické vedení a poskytnuté konzultace, jež značnou měrou přispěly k dokončení této práce. Věřím, že tyto předané zkušenosti a rady z oblasti návrhu elektroniky budou pro mne přínosem i nad rámec studijních povinností. Dále bych chtěl poděkovat Ing. Martinu Stankušovi, Ph.D za rady a ochotu, kterou projevil při konzultacích nastalých problémů v oblasti firmwaru.

## **Abstrakt**

Tato diplomová práce se zaměřuje na návrh a následnou realizaci komunikátoru, jakožto samostatného vestavěného elektronického zařízení. Ústřední funkcionalitou komunikátoru je sběr dat z IQRF MESH sítě a jejich následné odeslání do vzdáleného datového úložiště v podobě databáze. Zhotovené zařízení nabízí kromě servisních funkcionalit, také energeticky výhodný režim pro prodloužení výdrže akumulátoru. Dále se práce zabývá vytvořením servisní počítačové aplikace, jež může běžet na kterékoliv běžně používané platformě a upravovat konfiguraci komunikátoru. Poslední část je věnována realizaci datového úložiště pro měřená data.

**Klíčová slova:** IQRF, MESH, komunikátor, Wi-Fi, internet věcí, koncentrace dat, nízká spotřeba, akumulátor, MySQL, Qt, C, C++

## **Abstract**

This master thesis is focused on design and subsequent realization of a communicator as an automatic embedded electronic device. The main function of the communicator is to collect data from IQRF MESH net and its following sending to outlying data storage in the form of a database. Created device apart from service functions offers an energy-saving mode to extend battery life. Furthermore, the work deals with the development of a service computer application that can run on any commonly used platform and modify the configuration of the communicator. The last part is devoted to the realization of a data storage for measured data.

**Key Words:** IQRF, MESH, communicator, Wi-Fi, internet of things, data concentration. low power, battery, MySQL, Qt, C, C++

# Obsah

<b>Seznam použitých zkratk a symbolů</b>	<b>9</b>
<b>Seznam obrázků</b>	<b>13</b>
<b>Seznam tabulek</b>	<b>15</b>
<b>Seznam výpisů zdrojového kódu</b>	<b>16</b>
<b>1 Úvod</b>	<b>17</b>
<b>2 Wi-Fi technologie</b>	<b>18</b>
2.1 Režimy Wi-Fi sítí . . . . .	18
2.2 IEEE 802.11 . . . . .	20
2.3 Zabezpečení Wi-Fi sítě . . . . .	22
<b>3 IQRF technologie</b>	<b>26</b>
3.1 IQRF brány . . . . .	28
3.2 IQRF IDE . . . . .	29
<b>4 Rodina mikrokontrolérů NXP Kinetis L</b>	<b>30</b>
4.1 Kategorie v rodině Kinetis L . . . . .	31
4.2 Jádru ARM Cortex-M0+ . . . . .	32
<b>5 Návrh a realizace hardwaru IQRF - Wi-Fi komunikátoru</b>	<b>34</b>
5.1 Návrh komunikátoru . . . . .	34
5.2 Realizace HW komunikátoru . . . . .	38
5.3 Ovládací a indikační prvky . . . . .	47
5.4 Zhotovený komunikátor . . . . .	48
<b>6 Návrh a realizace firmwaru IQRF - Wi-Fi komunikátoru</b>	<b>50</b>
6.1 Ovladače . . . . .	50
6.2 Systém . . . . .	59
<b>7 Popis možností a realizace přenosu dat z IQRF sítě na datové úložiště</b>	<b>67</b>
7.1 Realizace datového úložiště . . . . .	68
<b>8 Návrh servisního komunikačního protokolu</b>	<b>71</b>
8.1 Návrh servisního protokolu . . . . .	71
8.2 Realizace servisní aplikace . . . . .	73

<b>9 Testování komunikátoru</b>	<b>75</b>
<b>10 Závěr</b>	<b>78</b>
<b>Literatura</b>	<b>79</b>
<b>Přílohy</b>	<b>80</b>
<b>A Schéma zapojení a deska plošných spojů</b>	<b>81</b>
<b>B Firmware</b>	<b>82</b>
<b>C Servisní aplikace</b>	<b>83</b>
<b>D PHP skripty a databáze</b>	<b>84</b>

## Seznam použitých zkratk a symbolů

AČ	– Analogově Číslicový
AES	– ( <i><b>A</b>dvanced <b>E</b>ncryption <b>S</b>tandard</i> ) - standardizovaný algoritmus pro šifrování dat
AP	– ( <i><b>A</b>ccess <b>P</b>oint</i> ) - přístupový bod
API	– ( <i><b>A</b>pplication <b>P</b>rogramming <b>I</b>nterface</i> ) - rozhraní pro programování aplikací
ARM	– ( <i><b>A</b>dvanced <b>R</b>ISC <b>M</b>achine</i> ) - označení architektury procesorových jader s RISC instrukční sadou
ASCII	– ( <i><b>A</b>merican <b>S</b>tandard <b>C</b>ode for <b>I</b>nformation <b>I</b>nterchange</i> ) - tabulka kódování znaků anglické abecedy
CCMP	– ( <i><b>C</b>ounter <b>C</b>ipher <b>M</b>ode with <b>B</b>lock <b>C</b>haining <b>M</b>essage <b>A</b>uthentication <b>C</b>ode <b>P</b>rotocol</i> ) - šifrovací protokol vyvinut pro účely bezdrátových sítí
COV	– ( <i><b>C</b>ell <b>O</b>ver<b>v</b>oltage</i> ) - přepětí článku
COC	– ( <i><b>C</b>ell <b>O</b>ver<b>c</b>urrent</i> ) - nadproud článku
CPU	– ( <i><b>C</b>entral <b>P</b>rocessing <b>U</b>nit</i> ) - centrální procesorová jednotka
CRC	– ( <i><b>C</b>yclic <b>R</b>edundancy <b>C</b>heck</i> ) - cyklický redundantní součet
CUV	– ( <i><b>C</b>ell <b>U</b>nder<b>v</b>oltage</i> ) - podpětí článku
ČA	– Číslicově Analogový
ČTÚ	– Český Telekomunikační Úřad
DC	– ( <i><b>D</b>irect <b>C</b>urrent</i> ) - stejnosměrný proud
DHCP	– ( <i><b>D</b>ynamic <b>H</b>ost <b>C</b>onfiguration <b>P</b>rotocol</i> ) - protokol z rodiny TCP/IP pro automatickou konfiguraci a přidělování IP adres koncovým zařízením
DIP	– ( <i><b>D</b>ual <b>I</b>n-line <b>P</b>ackage</i> ) - pouzdro pro elektronické komponenty typické svým obdélníkovým půdorysem a dvěma rovnoběžnými řadama kontaktovacích nožiček
DNS	– ( <i><b>D</b>omain <b>N</b>ame <b>S</b>ystem</i> ) - hierarchický systém doménových jmen
DPA	– ( <i><b>D</b>irect <b>P</b>eripheral <b>A</b>ccess</i> ) - komunikační protokol pro ovládání a správu zařízení v IQMESH sítích
DPS	– Deska Plošných Spojů
DSSS	– ( <i><b>D</b>irect <b>S</b>equence <b>S</b>pread <b>S</b>pectrum</i> ) - technika přímého rozprostření spektra
EAP	– ( <i><b>E</b>xtensible <b>A</b>uthentication <b>P</b>rotocol</i> ) - autentizační protokol
EMC	– ( <i><b>E</b>lectromagnetic <b>C</b>ompatib<b>i</b>lity</i> ) - elektromagnetická kompatibilita
ESD	– ( <i><b>E</b>lectrostatic <b>D</b>ischarge</i> ) - elektrostatický výboj

FRAM	– ( <i>Ferroelectric Random Access Memory</i> ) - feroelektrická energeticky nezávislá paměť s přímým přístupem
FUP	– ( <i>Fair User Policy</i> ) - označení pro limit přenesených dat po síti
GPIO	– ( <i>General-Purpose Input/Output</i> ) -
GSM	– ( <i>Groupe Spécial Mobile</i> ) - globální systém mobilní komunikace
GUI	– ( <i>Graphical User Interface</i> ) - grafické uživatelské rozhraní
HDLC	– ( <i>High-Level Data Link Control</i> ) - komunikační protokol linkové vrstvy
HTTP	– ( <i>Hypertext Transfer Protocol</i> )- protokol pro přenos hypertextových dokumentů
HWP	– <b>H</b> ardwarový profil
I <sup>2</sup> C	– ( <i>Inter-Integrated Circuit</i> ) - multi-masterová sériová sběrnice
IDE	– ( <i>Integrated Development Environment</i> ) - vývojové prostředí
IEEE	– ( <i>Institute of Electrical and Electronics</i> ) - institut pro elektrotechnické a elektronické inženýrství
IO	– <b>I</b> ntegrovaný <b>O</b> bvod
IoT	– ( <i>Internet of Things</i> ) - internet věcí
JTAG	– ( <i>Joint Test Action Group</i> ) - standard pro testování a ověřování DPS včetně ladění CPU
LDO	– ( <i>Low Dropout</i> ) - typicky regulátor, jež je schopný pracovat se vstupním napětím blízkým výstupnímu
LED	– ( <i>Light-Emitting Diode</i> ) - světlo vyzařující dioda
MAC	– ( <i>Media Access Control</i> ) - hardwarově závislá podvrstva linkové vrstvy v ISO/OSI modelu
MEMS	– <b>M</b> ikroelektromechanická součástka
MCU	– ( <i>Microcontroller Unit</i> ) - mikrokontrolér
MIC	– ( <i>Message Integrity Check</i> ) - kontrola integrity zprávy
MIMO	– ( <i>Multiple Input Multiple Output</i> ) - více vstupů více výstupů, abstraktní matematický model pro multi-anténní komunikační systémy
MOSFET	– ( <i>Metal Oxide Semiconductor Field Effect Transistor</i> ) - tranzistor jehož vodivost je řízena elektrickým polem ve struktuře kov-oxid-polovodič
MQTT	– ( <i>Message Queuing Telemetry Transport</i> ) - komunikační protokol pro zásílání zpráv, často využívaný v IoT
OEM	– ( <i>Original Equipment Manufacturer</i> ) - označení pro softwaru, který je kompatibilní jen s určitým typem hardwaru

OFDM	– ( <i>Orthogonal Frequency Division Multiplexing</i> ) - širokopásmová modulace, která využívá více ortogonálních frekvencí v rámci kanálu pro přenos dat
OTG	– ( <i>On-The-Go</i> ) - specifikace USB umožňující zařízení typu host převzít roli hostitele
PLL	– ( <i>Phase-Locked Loop</i> ) - fázový závěs
PPP	– ( <i>Point-to-Point Protocol</i> ) - komunikační protokol linkové vrstvy pro přímé spojení mezi dvěma uzly
PSK	– ( <i>Pre-Shared Key</i> ) - předsdílený klíč
QoS	– ( <i>Quality of Service</i> ) - služba pro řízení a rezervaci datových toků
RC	– ( <i>Resistor-Capacitor</i> ) - rezistor-kondenzátor
RC	– ( <i>Real Time Clock</i> ) - hodiny reálného času
RF	– <b>R</b> ádiové <b>F</b> rekvence
RISC	– ( <i>Reduced Instruction Set Computer</i> ) - označení pro redukovanou instrukční sadu
ROM	– ( <i>Read-Only Memory</i> ) - nepřepisovatelná elektronická paměť
RX	– ( <i>Receive</i> ) - přijímat
SCG	– ( <i>System Clock Generator</i> ) - modul pro ovládání rozvodu hodinového signálu
SDK	– ( <i>Software Development Kit</i> ) - sada vývojových nástrojů pro vytváření softwaru
SIM	– ( <i>Subscriber Identity Module</i> ) - účastnická identifikační karta
SMT	– ( <i>Surface Mount Technology</i> ) - povrchová montáž
SoC	– ( <i>System on Chip</i> ) - systém na čipu
SOC	– ( <i>State Of Charge</i> ) - stupeň nabití článku/akumulátoru
SOC	– ( <i>Secondary Oscillator</i> ) - druhotný oscilátor
SOH	– ( <i>State Of Health</i> ) - stupeň životnosti článku/akumulátoru
SPI	– ( <i>Serial Peripheral Interface</i> ) - sériové periferní rozhraní
SSID	– ( <i>Service Set Identifier</i> ) - identifikátor bezdrátové Wi-Fi sítě
SWD	– ( <i>Serial Wire Debug</i> ) - dvou vodičové ladící rozhraní pro ARM procesory
TCP	–
TKIP	– ( <i>Temporal Key Integrity Protocol</i> ) - bezpečnostní protokol součástí WPA zabezpečení Wi-Fi sítí
TX	– ( <i>Transmit</i> ) - odesílat
UART	– ( <i>Universal Asynchronous Receiver and Transmitter</i> ) - asynchronní sériové rozhraní
USB	– ( <i>Universal Serial Bus</i> ) - univerzální seriová sběrnice

VoIP	– ( <b>V</b> oice <b>o</b> ver <b>I</b> nternet <b>P</b> rotocol) - protokol určenou pro internetovou telefonii
WEP	– ( <b>W</b> ired <b>E</b> quivalent <b>P</b> rivacy) - označení pro zabezpečení bezdrátových sítí
WiGig	– ( <b>W</b> ireless <b>G</b> igabit <b>A</b> lliance) -
WPA	– ( <b>W</b> i- <b>F</b> i <b>P</b> rotected <b>A</b> ccess) - obchodní označení pro zabezpečení bezdrátových sítí
WYSIWYG	– ( <b>W</b> hat <b>Y</b> ou <b>S</b> ee <b>I</b> s <b>W</b> hat <b>Y</b> ou <b>G</b> et) - druh editorů, kde se editovaný obsah upravuje graficky, tak jak vypadá jeho finální podoba



## Seznam obrázků

1	Znázornění Wi-Fi sítě v režimu ad hoc . . . . .	19
2	Znázornění Wi-Fi sítě v infrastrukturním režimu . . . . .	20
3	Schématické znázornění WEP šifrování . . . . .	24
4	Schématické znázornění fungování IEEE 802.1X . . . . .	25
5	IQRF vývojová sada DS-START-04 . . . . .	26
6	Znázornění struktury programování jen s využitím OS[13] . . . . .	27
7	Znázornění použití jen s nahraným HWP[13] . . . . .	27
8	Znázornění struktury programování s využitím DPA[13] . . . . .	28
9	Ukázka GUI od IQRF IDE . . . . .	29
10	Přehled podskupin rodiny Kinetis <sup>®</sup> [14] . . . . .	30
11	Vnitřní struktura jádra ARM Cortex-M0+[4] . . . . .	33
12	Topologie použití komunikátoru . . . . .	34
13	Blokové schéma hlavních částí komunikátoru . . . . .	34
14	Blokový diagram struktury KL28Z512VLL7[4] . . . . .	35
15	Vybraný Wi-Fi modul RN-131 [10] . . . . .	36
16	Blokový diagram Wi-Fi modulu RN-131 [10] . . . . .	37
17	Blokový diagram DCTR-76D[15] . . . . .	38
18	Vytvořený akumulátor pro napájení komunikátoru . . . . .	39
19	Snímek obrazovky Battery Management Studia . . . . .	40
20	Schéma zapojení integrovaných obvodů pro ochranu akumulátoru . . . . .	41
21	Schéma zapojení stabilizátoru na 3.3 V . . . . .	42
22	Schéma zapojení nabíječe akumulátoru . . . . .	42
23	Schéma zapojení napájecích pinů MCU . . . . .	43
24	Schéma připojení USB rozhraní . . . . .	44
25	Schéma připojení SWD ladícího rozhraní . . . . .	44
26	Schéma připojení rezonačního krystalu k MCU . . . . .	45
27	Schéma zapojení externí FRAM paměti CY15B104Q-LHXI . . . . .	45
28	Schéma zapojení Wi-Fi modulu . . . . .	46
29	Schéma zapojení IQRF modulu DCTR-76D . . . . .	47
30	Schéma zapojení programovací konektoru a napájení pro IQRF modul . . . . .	47
31	Osazená DPS navrženého komunikátoru . . . . .	49
32	Zhotová DPS uvnitř vybrané krabičky . . . . .	49
33	Stavový diagram systému komunikátoru . . . . .	60
34	UML diagram aktivit pro režim IQRF tunelu . . . . .	61
35	UML diagram aktivit pro režim Wi-Fi tunelu . . . . .	62
36	Zjednodušený UML diagram aktivit pro běžný režim . . . . .	64
37	Zjednodušený UML diagram aktivit pro režim nízké spotřeby . . . . .	66

38	Schématické znázornění přenosu dat z IQMESH sítě pomocí IQRF brány . . . .	67
39	Diagram navržené databáze . . . . .	69
40	Ukazáka výpisu tabulky MySQL databáze . . . . .	70
41	Zobrazení struktury paketu . . . . .	71
42	Zobrazení dvou možných scénářů komunikace . . . . .	71
43	Ukázka vývojového prostředí Qt Creator . . . . .	73
44	UML diagram případu užití pro vytvořenou aplikaci . . . . .	74
45	Znázornění vytvořené testovací sítě . . . . .	75
46	Scháma zapojení použité při měření spotřeby . . . . .	76
47	Časový průběh spotřeby při sběru dat z IQRF v úsporném režimu . . . . .	77
48	Časový průběh spotřeby při sběru a odesílání dat v úsporném režimu . . . . .	77

## Seznam tabulek

1	Technické parametry pro širokopásmový přenos dat v pásmech 2,4 GHz až 66 GHz	18
2	Srovnání aktuálně doporučovaných IQRF modulů . . . . .	37
3	Tabulka změřených hodnot spotřeby . . . . .	76

## Seznam výpisů zdrojového kódu

1	Veřejné prototypy ovladače RTC . . . . .	51
2	Veřejné prototypy knihovny Queue . . . . .	52
3	Veřejné prototypy ovladače LPUART . . . . .	53
4	Veřejné prototypy ovladače Wi-Fi modulu . . . . .	56
5	Veřejné prototypy ovladače IQRF modulu . . . . .	57
6	Veřejné prototypy ovladače FRAM paměti . . . . .	58
7	Veřejné prototypy ovladače bateriového monitoru . . . . .	58
8	Veřejné prototypy ovladače USB . . . . .	59
9	Datové typy použité pro komunikační protokol . . . . .	72

# 1 Úvod

V dnešních dnech, neustávající technického pokroku lidstva a nástupem čtvrté průmyslové revoluce, je čím dál více zařízení vybavováno integrovanou elektronikou. Tento proud se netýká jen průmyslového vybavení, ale i věcí každodenní potřeby. Především evolučním krokem bylo všechna zařízení automatizovat. Takzvaný internet věcí (IoT), však tuto myšlenku posílá ještě dále a představuje svět, kde jsou všechna zařízení spolu propojená a navzájem komunikují. Samostatné odvětví pak tvoří bezdrátové komunikační technologie, jenž tato jednotlivá zařízení propojují. Mezi tyto technologie můžeme bezesporu zařadit i produkty jičínské společnosti Microrisc, s.r.o., která stojí za vznikem produktů pod značkou IQRF, a jež byly využity v této diplomové práci. Výhodou IQRF technologie je možnost vytvoření sítí se smíšenou MESH topologií, jejímž přínosem je zvýšení dosahu a dostupnosti připojení. Důležitou oblastí je sběr dat z těchto sítí s následným odesláním na datové úložiště. K tomuto účelu Microrisc nabízí IQRF brány, které však nejsou navrženy s ohledem na spotřebu elektrické energie. Tato skutečnost je limitující pro použití mimo dosah stabilního napájení. Cílem této diplomové práce je vytvořit zařízení, které by bylo i za těchto okolností použitelné. Pro připojení komunikátoru k datovému úložišti je zvolena Wi-Fi, která v dnešní době patří k jednomu z nejrozšířenějších komunikačních standardů pro připojení k internetu, zvláště v obydlených oblastech.

Na seznámení s Wi-Fi technologií je vyčleněna kapitola druhá. V ní jsou podrobně rozebrány možnosti tvorby Wi-Fi, které historicky používaly různé standardy. Spolu s tímto jsou také rozepsány možné režimy fungování a způsoby zabezpečení. Třetí kapitola je věnována popisu technologie IQRF. Po představení, jsou uvedeny nejdůležitější informace a také popsány softwarové a hardwarové prostředky. Popisu a představení rodiny mikrokontroléru NXP Kinetis® L je věnována kapitola čtvrtá. V ní je přehled jednotlivých podkategorií, předurčení aplikací a společné parametry. První částí, která se již nezabývá jen teoretickým rozбором je pátá kapitola. Ta popisuje výběr ústředních komponent, spolu s uvedením parametrů a také návrh hardwaru komunikátoru. V samotném závěru kapitoly je popsán zhotovený prototyp. Kapitola šestá má za cíl rozebrat vznik firmwaru komunikátoru. Jsou zde popsány jednotlivé programové celky od ovladačů periferií, až po samotný aplikační firmware s různými režimy běhu. Další kapitola se ve své první půli zaměřuje na existující možnosti přenosu dat z IQMESH sítě na cloudové úložiště. Druhá část již popisuje uskutečněné řešení serverové části. V osmé kapitole je představen navržený komunikační protokol pro konfigurační zásahy do komunikátoru, spolu s popisem realizované počítačové aplikace, jež tento protokol implementuje. Poslední část práce je zaměřena na testování zhotoveného komunikátoru s ohledem na spotřebu elektrické energie. Je zde popsán způsob testování a porovnání výsledků běžného a úsporného režimu.

## 2 Wi-Fi technologie

Samotný název Wi-Fi, či WiFi, někdy mylně vykládán jako zkratka z anglického *wireless fidelity* (bezdrátová věrnost), je pojem, který v informatice označuje bezdrátovou komunikaci v počítačových sítích, jež je definována několika standardy IEEE 802.11. Pro provoz jsou využívána tzv. bezlicenční pásma 2,4 GHz, respektive 5 GHz, u kterých není nutné žádat o povolení a platit licenční poplatek. Přesto Český telekomunikační úřad upravuje podmínky, za kterých lze vysílání provozovat (Směrnice ČTU č. VO-R/12/09.2010-12 ze dne 29. září 2010).

Tabulka 1: Technické parametry pro širokopásmový přenos dat v pásmech 2,4 GHz až 66 GHz

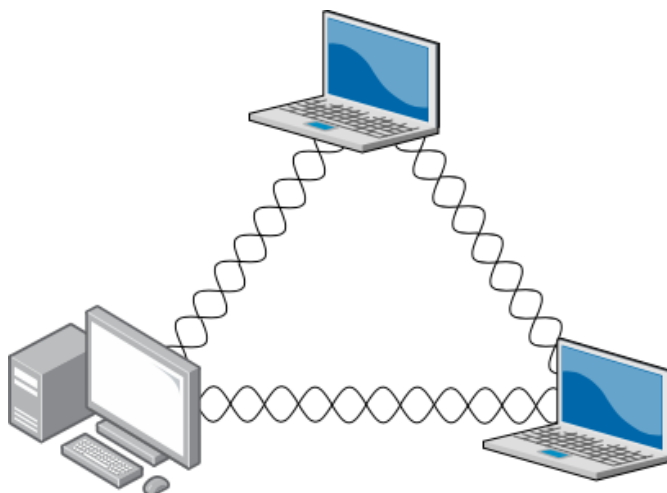
Ozn.	Kmitočtové pásmo	Vyzářený výkon	Maximální spektrální hustota e.i.r.p.	Další podmínky
a	2400,0 - 2483,5 MHz	100 mW e.i.r.p.	10 mW/1 MHz 100 mW/100 kHz	systémy s technikou DSSS nebo OFDM systémy s technikou FHSS
b	5150 - 5250 MHz	200 mW střední e.i.r.p.	10 mW/MHz (střední spektrální hustota v libovolném úseku širokém 1 MHz)	pouze pro použití uvnitř budovy
c	5250 - 5350 MHz	200 mW střední e.i.r.p.	10 mW/MHz (střední spektrální hustota v libovolném úseku širokém 1 MHz)	pouze pro použití uvnitř budovy
d	5470 - 5725 MHz	1 W střední e.i.r.p.	50 mW/MHz (střední spektrální hustota v libovolném úseku širokém 1 MHz)	—
e	17,1 - 17,3 GHz	100 mW střední e.i.r.p.	—	—
f	57 - 66 GHz	40 dBm střední e.i.r.p.	13 dBm/MHz (střední spektrální hustota)	Stálé venkovní instalace jsou vyloučeny

### 2.1 Režimy Wi-Fi sítí

Ačkoliv může být Wi-Fi síť vytvářena různými způsoby s rozličnou rozsáhlostí, vychází vždy z požadavků na konkrétní účel. Společným znakem je přítomnost identifikátoru SSID (Service Set Identifier). Jedná se o textový řetězec s maximální délkou 32 ASCII znaků, jež je pravidelně všesměrově vysílán (tzv. *broadcast*) a rozlišuje danou síť. Všechny Wi-Fi sítě lze dále rozdělit do dvou skupin, podle režimu, ve kterém pracují, a to buď ad hoc, či infrastrukturní.

### 2.1.1 Režim ad hoc

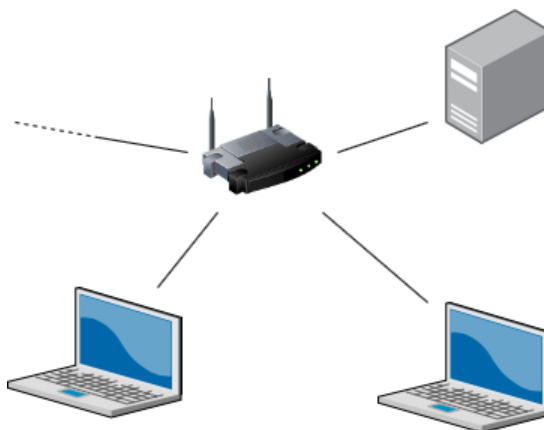
Jedná se o režim bezdrátové sítě, který je decentralizovaný. Není tudíž závislý na žádné existující infrastruktuře – neobsahuje žádné přístupové body (AP) a jednotliví klienti jsou v rovnocenné pozici tzv. *peer to peer*. Jednotlivé koncové zařízení spolu komunikují přímo, bez prostředníka. Jeho nesporná výhoda spočívá především v mobilitě a snadnosti vytvoření příležitostné sítě na malých vzdálenostech. V rámci členitějších a rozsáhlých sítí však uplatnění již nenachází, jelikož veškerá zařízení, která spolu chtějí komunikovat musí být ve vzájemném dosahu, přičemž je omezena maximální rychlost komunikace na 11 Mbit/s. [9]



Obrázek 1: Znázornění Wi-Fi sítě v režimu ad hoc

### 2.1.2 Režim infrastruktury

V případě tohoto režimu existuje přesně vymezená infrastruktura, jež se skládá z jednoho nebo několika přístupových bodů (AP). Ty vysílají své SSID, pomocí něhož vymezují svou síťovou příslušnost. Tento jediný přístupový bod dokáže obhospodařovat více klientů, kteří tak dokážou komunikovat mezi sebou, buď přes prostředníka, nebo se zařízením, které je připojeno k AP metalickým, či optickým vedením. Více přístupových bodů může mít stejný identifikátor a klient se zpravidla připojuje na ten s nejlepším dosahem. Ve větších sítích lze realizovat přepojování klienta mezi jednotlivými AP v závislosti na úrovni signálu. To umožní volné pohybování na velké oblasti bez ztráty připojení (tzv. *roaming*).



Obrázek 2: Znáznornění Wi-Fi sítě v infrastrukturním režimu

## 2.2 IEEE 802.11

V roce 1997 vznikl původní standard 802.11, jež definoval způsob komunikace bezdrátových sítí v pásmu 2,4 GHz. Podporovaná přenosová rychlost 1-2 Mb/s však záhy přestala stačit potřebám rozmáhajícímu se využití počítačových sítí, což vedlo k vydání prvních dvou vysokorychlostních dodatků 802.11a a 802.11b.

- **IEEE 802.11a** - Dodatek, který byl vydán v roce 1999 a definuje komunikaci v 5GHz pásmu. S použitou OFDM modulací je teoretická přenosová rychlost 54 Mbit/s. Povoluje vyšší vysílací výkon než je tomu v případě 802.11b/g. Výrobky splňující tento standard se v České republice zpravidla nenabízejí, jelikož jsou primárně určeny pro trhy v USA, Japonsku a odporují české legislativě.
- **IEEE 802.11b** - Tento standard od roku 1999 doplňuje původní normu IEEE 802.11 pro komunikaci v přenosovém pásmu 2,4 GHz. Díky tomu byla, oproti předchůdci, zvednuta maximální přenosová rychlost na 11 Mbit/s a teoretický dosah ve volném prostředí na 12 km.
- **IEEE 802.11c** - Doplňuje standard IEEE 802.11D a definuje přemostování v bezdrátových zařízeních. Především popisuje procedury v rámci MAC podvrstvy síťových mostů. (2001)
- **IEEE 802.11d** - Někdy také nazýván jako globální harmonizační standard je uplatňován především v zemích, kde nejsou povolena zařízení vyhovující jiným dodatkům k IEEE 802.11. Jedná se především o požadavky na fyzickou vrstvu, konkrétně vyhovění regulačním pravidlům, která nejsou pokryta ostatními standardy. Rozdíl lze nalézt v povolených frekvencích, propustnosti signálu a maximálních přípustných vyřazovacích výkonech. (2001)
- **IEEE 802.11e** - V případě tohoto doplňku je rozšiřována MAC podvrstva linkové vrstvy o podporu QoS. Nesmírnou důležitost tohoto standardu lze najít v aplikacích, které nepří-



znivě ovlivňuje jakékoliv zpoždění. Mezi ně se bezesporu řadí internetová telefonie VoIP, či proudy multimediálních dat. (2005)

- **IEEE 802.11F** - V roce 2003 vstoupil v platnost standard, který doporučuje jak má probíhat komunikace a spolupráce mezi přístupovými body (IAPP). Z kraje roku 2006 však došlo výkonným výborem IEEE 802 ke stažení.
- **IEEE 802.11g** - Roku 2003 taktéž došlo k rozšíření dosavadního IEEE 802.11b, kde byla, při zachování kmitočtu 2400 - 2485 MHz, navýšena teoretická přenosová rychlost na 54 Mbit/s. Zároveň však byla zachována zpětná kompatibilita, jelikož modulační schéma OFDM pro rychlosti 1, 2, 5,5 a 11 Mbit/s je totožné jako v předchozí generaci.
- **IEEE 802.11h** - Po 5 letech od vydání IEEE 802.11a vychází doplnění, které je navrženo pro provoz v evropských podmínkách i mimo budovy. Jelikož na kmitočtu 5 GHz pracují další zařízení jako jsou meteorologické radary, nebo některé druhy satelitních systémů, bylo potřeba vyřešit problémy související s vzájemným rušením. Výsledkem je integrace dynamického výběru kanálu do MAC podvrstvy a také mechanismus, který v případě detekce rušení na bezdrátovém zařízení omezí vysílací výkon, nebo uvolní inkriminovaný kanál.
- **IEEE 802.11i** - Tento dodatek z roku 2004 posiluje bezpečnost díky vylepšení autentizačního a šifrovacího algoritmu WPA2.
- **IEEE 802.11j** - Standard, který alokuje nové frekvenční rozsahy pro multimediální služby na území Japonska. (2004)
- **IEEE 802.11k** - Navazuje na IEEE 802.11j, kde pro vysoké frekvence vylepšuje správu radiových zdrojů.
- **IEEE 802.11n** - V tomto standardu dochází k úpravě MAC podvrstvy s cílem zvýšit propustnost sítě. Navýšení rychlosti je realizováno technologií MIMO (*multiple input multiple output*). Ta je založena na využití vícero vysílačů i přijímačů na jednotlivých zařízeních. Díky tomu může být maximální fyzická rychlost až 600 Mbit/s.
- **IEEE 802.11p** - Popisuje bezdrátový přístup pro pohyblivé prostředí, jako jsou auta, či vlaky.
- **IEEE 802.11r** - Roamingový standard - definuje časté a rychlé střídání přístupových bodů bezdrátovým zařízením.
- **IEEE 802.11s** - Následující rozšíření pojednává o principech tzv. *ESS Mesh Networking* - samoorganizující se bezdrátové sítě.
- **IEEE 802.11u** - Další z řady norem se zabývá spoluprací zařízení s mobilními a jinými bezdrátovými sítěmi, které nespadají do standardu 802.

- **IEEE 802.11v** - Dodatek, který upravuje možnosti správy klientských zařízení v bezdrátových sítích během jejich připojení.
- **IEEE 802.11y** - Tento standard specifikuje provoz bezdrátových zařízení ve frekvenčním pásmu 3650 - 3700 MHz, které je ve Spojených státech amerických nelicencované. Teoretický dosah ve volném prostředí je až 5 km.
- **IEEE 802.11ac** - Nový standard, který vylepšuje a rozšiřuje svého předchůdce IEEE 802.11n. Teoretická maximální rychlost je navýšena na 1 Gbit/s díky využívání jak 2,4 GHz, tak 5 GHz pásma, možnost využít širší RF pásmo (do 160 MHz), zvětšení počtu MIMO kanálů (až 8) a v neposlední řadě použití modulace s vyšší hustotou (až 256-QAM).
- **IEEE 802.11ad** - Obchodní sdružení WiGig (Wireless Gigabit Alliance) vzniklo v roce 2009 k podpoře normy IEEE 802.11ad. Jedná se o multi-gigabitovou bezdrátovou technologii, která pro svou činnost využívá nejen kmitočty 2,4 GHz a 5 GHz, ale také bezlicenční pásmo 60 GHz. Tímto je umožněna bezdrátová komunikace o maximální rychlosti 7 Gbit/s. Signál o tak vysoké frekvenci již ovšem zpravidla neproniká stěnami. Z toho důvodu je standard zpětně kompatibilní a nastane-li taková situace může zařízení komunikovat na nižších kmitočtech. Zkraje roku 2013 oznámila Wireless Gigabit Alliance odtržení od Wi-Fi Alliance a později vydala prohlášení, že protokol bude použit v nové verzi USB.

## 2.3 Zabezpečení Wi-Fi sítě

Palčivou otázkou u Wi-Fi sítí je problém zabezpečení. Zatímco u metalického/optického vedení je pro odposlech, či nabourání se do komunikace potřeba fyzického připojení přímo na vedení, u Wi-Fi mnohdy stačí útočníkovi použít směrovou anténu a dokáže se připojit i na vzdálenost několika stovek metrů. To je zapříčiněno tím, že se rádiové vlny šíří všemi směry, a to i skrz hmotu. Proto je velmi obtížné omezit prostor, kde je lze zachytit.

Důležitým tématem, se zabezpečení stává především tam, kde proudí citlivá data, a to jako osobního, tak obchodního charakteru. Postupem času se vyvíjely různé metody zabezpečení od těch jednoduchých po sofistikované. Ideální volbou je proto kombinace více zabezpečovacích technik. Starší zařízení však může limitovat nedostatečný výkon pro pokročilé šifrování (WPA2). Bezpečnost lze obecně rozdělit do dvou skupin:

- Autorizace - zabránění přístupu neoprávněným uživatelům.
- Šifrování - zabezpečení před odposlechem, pomocí zakódování přenášených dat.

### 2.3.1 RF signálové stínění

Spíše za doplňkovou ochranu lze považovat omezení dosahu rádiového signálu. Toho lze docílit vhodným umístěním přístupových bodů, případně snížením vyzařovaného výkonu, či použitím

směrových nebo úsekových antén. Další možnost, jak zamezit šíření rádiových vln mimo budovu je použít specializované barvy na zdi a okenní fólie, které výrazně oslabují bezdrátový signál.

### 2.3.2 Skrytí SSID

Jak je zmíněno výše, každý přístupový bod pravidelně vysílá tzv. *beacon* pakety, pomocí nichž signalizuje okolním zařízením svou přítomnost. Tyto zprávy však obsahují mimo další informace také SSID. Zablokováním vysílání je sice porušen standard, ale uživatele poté tuto síť běžně nevidí. K připojení je pak nutné znát SSID, jinak nemůže dojít k asociaci. Toto snižuje riziko připojení nechtěné stanice, avšak pro případného útočníka nepředstavuje velký problém. Pro zjištění SSID je možno pomocí softwaru, jako je *aircrack-ng*, do sítě zaslat falešnou zprávu o odpojení připojeného klienta od AP a poté odposlouchávat komunikaci. Odpojený klient při pokusu o připojení vysílá SSID v nešifrované formě.

### 2.3.3 Filtrování MAC adres

Mezi další jednodušší techniky, které zabraňují připojení neoprávněným uživatelům je nástroj na AP, který přijímá nebo odmítá asociaci klienta podle jeho MAC adresy. Tato adresa je jedinečná, pevně stanovena výrobcem Wi-Fi modulu/karty a je zapsána v paměti typu ROM. Používá se pro identifikaci zařízení a směrování v rámci jedné sítě na úrovni linkové vrstvy (2. nejnížší v referenčním modelu ISO/OSI). V případě provozu Wi-Fi pod operačním systémem, je zpravidla tato adresa při spouštění načtena do operační paměti, kde se s ní nadále pracuje. Následným odposlechem lze z hlavičky přijatých rámců separovat MAC klienta, jenž je k AP připojen. Tuto fyzickou adresu pak jen stačí přepsat v operační paměti a zařízení se bude pro okolní systém vydávat za jiného klienta, který se nachází na "*white listu*".

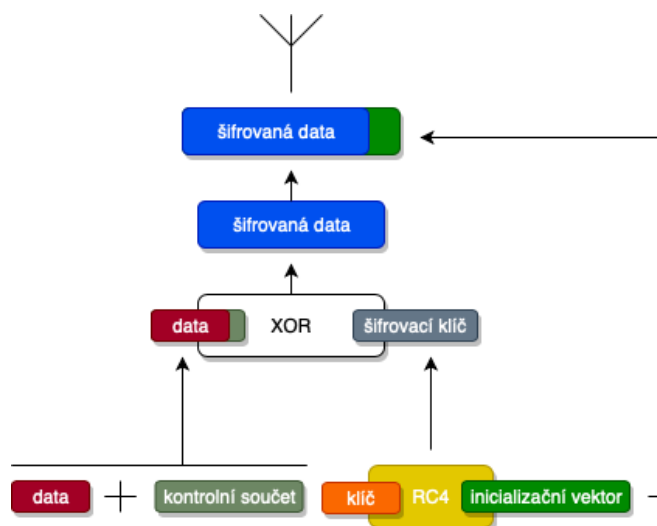
### 2.3.4 Vypnutí DHCP služby

Při použití této metody dochází k vypnutí služby DHCP (obvykle běží na routeru, v rozsáhlých sítích na serveru), která se stará o dynamické přidělování logických adres jednotlivým klientům. Spolu se samotnou klientskou IP adresou, předává také síťovou masku (jež odděluje hostovou část od síťové), adresu výchozí brány a rovněž adresu DNS serveru (překlad doménových názvů na IP adresy). K připojení do sítě bez aktivního DHCP je nutné všechny tyto údaje znát, jinak směrování paketů, případně překlad doménových názvů nebude funkční.

### 2.3.5 WEP

WEP protokol je obsažen v standardu IEEE 802.11 od roku 1999. Vznikl za účelem zabezpečení bezdrátové komunikace stejně tak, jako to zajišťuje izolace u drátového spojení. Uživatele v rámci sítě vůči sobě však chránění nejsou. WEP využívá symetrickou šifru RC4 pro zabezpečení a kontrolní součet CRC-32 pro integritu. Používají se 64, 128 a 256bitové klíče, inicializační vektor má 24 bitů. O 2 roky později po uvedení bylo WEP šifrování prolomeno. Díky dířám v protokolu

lze zachytáváním a následnou analýzou specifických rámců klíč pomocí specializovaného SW zjistit (je obsažen v balíku *aircrack-ng*). Ačkoliv je tato chyba dlouho známa, stále lze i nyní narazit na sítě s tímto zabezpečením. Podnikové nasazení však stěžuje absence správy klíčů, je nutné mít pro všechny klienty jedno jediné heslo a o jeho případně změně musí být informováni všichni uživatelé.



Obrázek 3: Schématické znázornění WEP šifrování

### 2.3.6 WPA

Na prolomení WEP šifrování reaguje v roce 2002 *Wi-Fi Alliance* vydáním popisu zabezpečení WPA (Wi-Fi Protected Access), které bylo součástí tehdy připravovaného standardu IEEE 802.11i. Vyvinutý protokol TKIP (Temporary Key Integrity Protocol) používá stejný šifrovací algoritmus jako WEP, avšak obsahuje dynamické generování dočasných klíčů každých 10000 paketů. Používá se 128bitový klíč v kombinaci s 48bitovým inicializačním vektorem. Pro zachování integrity slouží algoritmus MIC, který obsahuje počítadlo rámců. To mimo jiné zajišťuje ochranu před útoky, které se snaží zopakovat předchozí komunikaci.

Autentizace klienta uvnitř sítě chráněné WPA protokolem může probíhat pomocí předsdílené fráze - PSK (Pre Shared Key) nebo pomocí autentizačního serveru - dle standardu IEEE 802.1X.

I přes prodloužení inicializačních vektorů, omezení zaslaných paketů s příbuzným klíčem a ověřování integrity zpráv, je dnes zabezpečené WPA řazeno na stejnou úroveň jako WEP a nedoporučuje se jej používat. Navíc pro sítě v režimu ad hoc se toto šifrování, stejně jako WPA2, nedá použít.

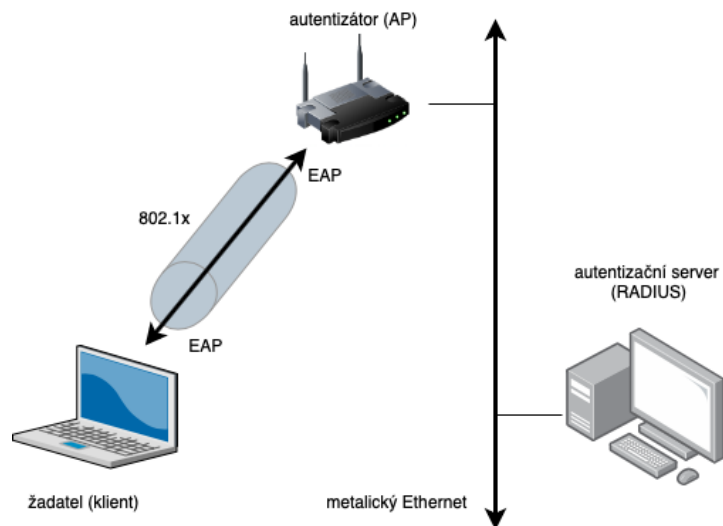
### 2.3.7 WPA2

WPA2 je zkrácený název pro implementaci dokončeného standardu 802.11i, který přináší kvalitnější šifrování a zlepšuje autentizační algoritmus, než tomu bylo u nevyhovujícího WPA. Běžně používaným zabezpečovacím protokolem je CCMP (Counter-Mode/Cipher Block Chaining Encryption Standard) založeném na šifrování AES (Advanced Encryption Standard). Stejně jako v předchozím zabezpečení je zde implementován MIC pro kontrolu integrity a ochrana proti opakovaným útokům. Důvodem proč na starších zařízeních není možné používat WPA2 zabezpečení je znatelně vyšší nárok na výpočetní výkon pro šifrování než tomu bylo u WPA.

### 2.3.8 IEEE 802.1X

Tento protokol, který byl původně zamýšlený pro kabelové sítě, se velmi dobře uchytil v oblasti zlepšení bezpečnosti bezdrátových sítí. Standard vychází z protokolu PPP v němž se lze autentizovat pouze na základě správně zadané kombinace jména a hesla. Avšak technologie IEEE 802.1X k samotné autentizaci využívá protokol EAP. Díky němu je umožněna autentizace různými způsoby. Využity mohou být hesla (PEAP-MSCHAPv2), certifikáty (EAP-TLS), čipové karty, tokeny apod.

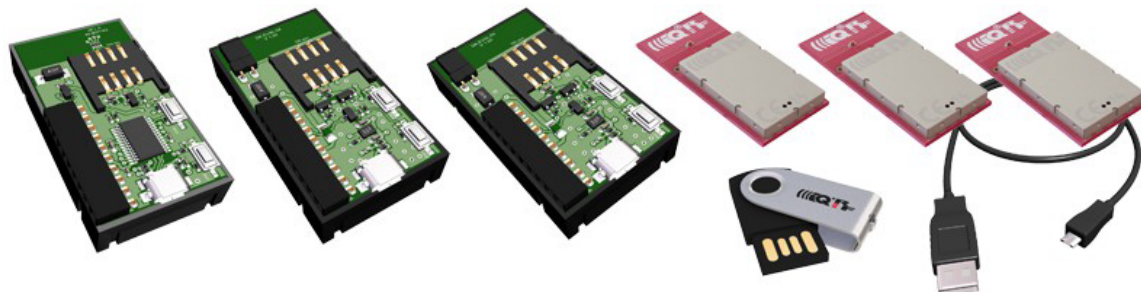
Pokud se chce nový klient připojit do Wi-Fi sítě, přístupový bod blokuje veškerou komunikaci, s výjimkou protokolu EAP, pomocí něj z klientského zařízení tzv. "prosebník" (suplikant) odešle na autentizační informace na RADIUS server. Server žádost vyhodnotí, provede ověření uživatele a na základě výsledku informuje přístupový bod, ke kterému byl klient připojen, zda má odblokovat ostatní komunikaci, či nikoliv.



Obrázek 4: Schématické znázornění fungování IEEE 802.1X

### 3 IQRF technologie

Historie inteligentní bezdrátové technologie sahá do roku 2004, kdy je začala produkovat česká firma IQRF Tech s.r.o. se sídlem v Jičíně.

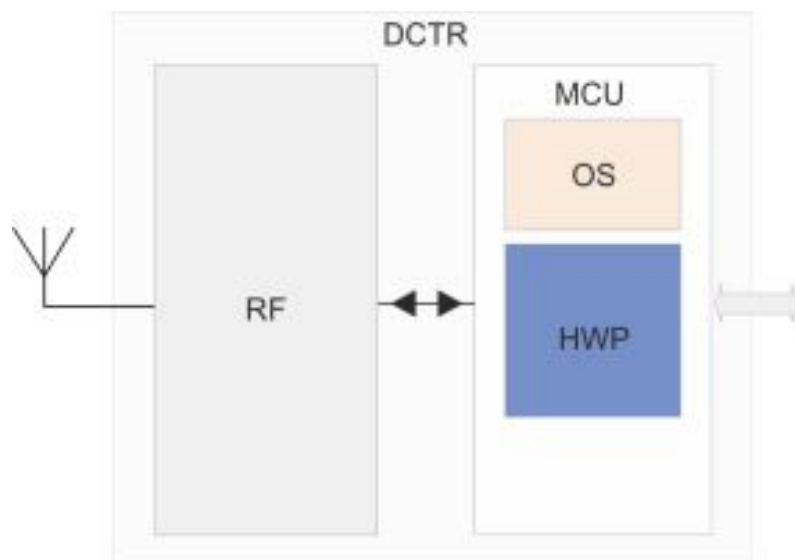


Obrázek 5: IQRF vývojová sada DS-START-04

Tato platforma je schopna bezdrátového přenosu menších objemů dat v rozsahu desítek až stovek metrů. Nízká rychlost přenosu dat (19,836 kb/s) je vyrovnávána velmi nízkým proudovým odběrem. Ten se liší na základě použitého režimu. (Sleep mód, Run mód, RX a TX mód, který je závislý na vysílacím výkonu). Nespornou výhodou IQRF je jeho snadná implementace a variabilita použití s jakýmkoliv elektrickým zařízením, které vyžaduje bezdrátový přenos. V Evropě pro přenos dat je využíváno frekvenční pásmo o frekvenci 868 MHz, kdy je možno využít až 62 kanálů, v USA se používá vysílací pásmo 916 MHz, celosvětově pak IQRF pracuje ve vysokofrekvenčním pásmu 433 MHz.

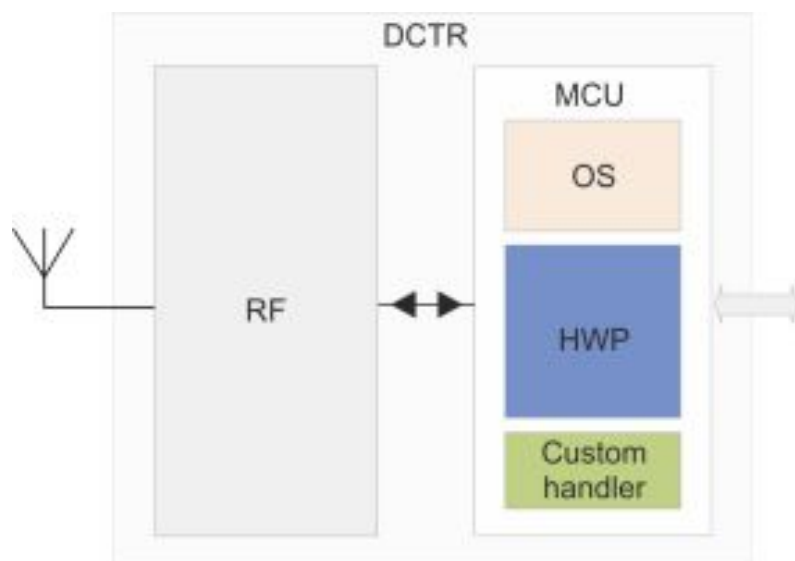
Základem platformy je RF modul pro obousměrný přenos dat s operačním systémem, který je vždy již součástí IQRF jako první vrstva. Napájecí napětí RF modulu - přijímače/vysílače se pohybuje mezi 3 až 5,3 V, kdy závisí na konkrétním typu použitého modulu.

Druhou vrstvu tvoří DPA (Direct Peripheral Access) protokol jehož součástí jsou tzv. HWP (Hardware Profiles) plug-in soubory. Pomocí tohoto protokolu, který je dodáván výrobcem, je možno nastavit hlavní přijímač/vysílač, který řídí síť IQRF a je označován jako koordinátor. Protokol DPA umožňuje také nastavení tzv. nodů – další přijímače/vysílače v síti, se kterými koordinátor synchronně komunikuje. Použitím DPA lze do sítě zapojit až 239 nodů.



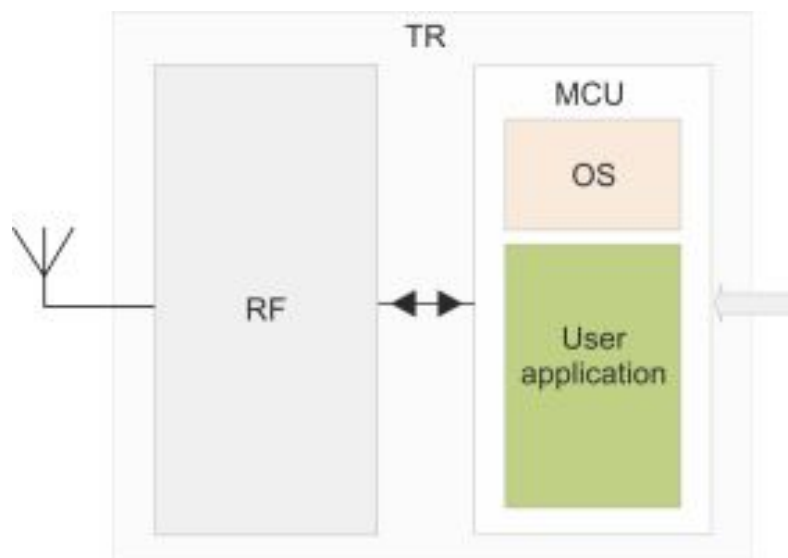
Obrázek 6: Znázornění struktury programování jen s využitím OS[13]

Třetí, poslední vrstvu může tvořit tzv. Custom DPA Handler. Tato aplikace souží jako nástavba, kdy hardwarové profily nejsou dostatečné. Díky této aplikaci je možno přizpůsobit chování vysílačů/přijímačů podle aktuálních potřeb uživatele.



Obrázek 7: Znázornění použití jen s nahraným HWP[13]

Tyto dvě vrstvy spadají pod DPA protokol a jsou jedinou možností pro využití IQMESH sítě. Druhou možností přístupu je vytvoření vlastní uživatelské aplikace v programovacím jazyce C, která podle vlastní požadavků ovládá funkce IQRf. Tento přístup již není podporován IQMESH sítí.



Obrázek 8: Znázornění struktury programování s využitím DPA[13]

Z výše uvedeného textu vyplývá, že existují pro platformu IQRF dva režimy sítí. První, již zmíněná IQMESH (MESH síť implementovaná na platformu IQRF) síť nemá žádné konkrétní uspořádání přijímačů/vysílačů, jak je například zvykem u hvězdicových či lineárních sítí. Jestliže se adresát nachází mimo přímý dosah, může mu být datový paket, který může obsahovat až 64 bajtů dat, doručen prostřednictvím jiných prvků sítě (routeru), které jsou v dosahu. Výhodou IQMESH sítě je, že její prvky mohou pracovat paralelně ve více nezávislých sítích. Druhý režim sítě je tzv. Peer to peer. Ten funguje na principu stejné úrovně všech prvků, kdy jsou datové pakety dostupné všem prvkům v dosahu. Na základě uživatelské aplikace jsou pak tyto pakety zpracovávány.

### 3.1 IQRF brány

Všeobecně brány (gateways) umožňují propojení dvou sítí, které pracují s různými komunikačními protokoly.

Využitím pouze přijímače/vysílače lze IQRF propojit s jednoduchými standardními rozhraními jako SPI, I2C, UART. Komplexnější standardy jako USB, Ethernet, GSM a WIFI již vyžadují přístup skrz bránu. Uživatel má také možnost přistupovat k síti IQRF přes vizuální ovládací panely, jež slouží jako přístupové brány.

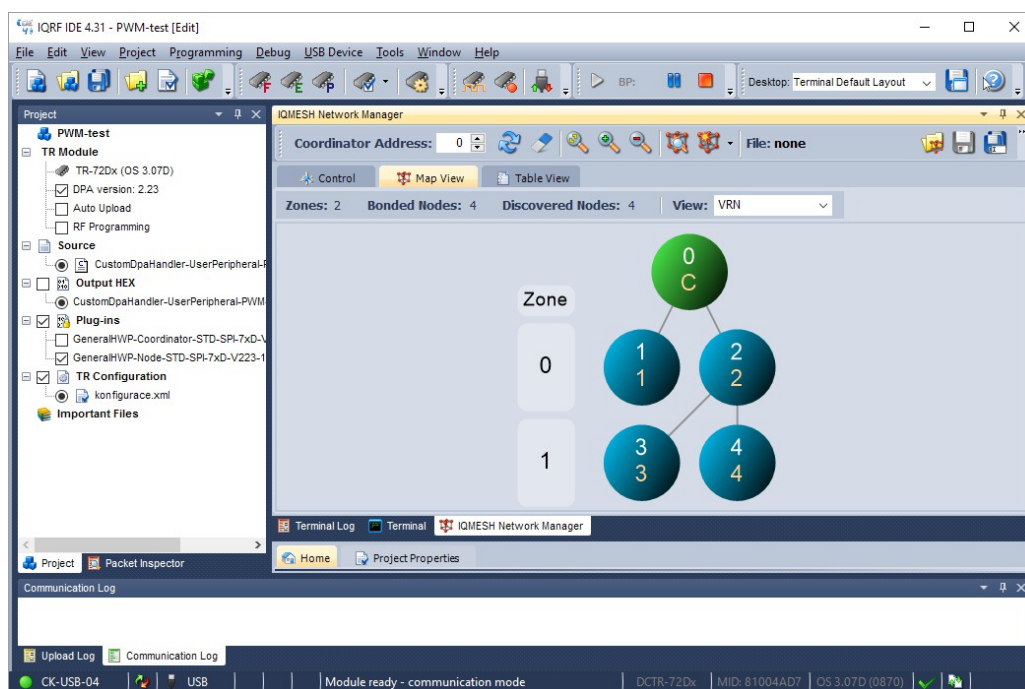
Brány IQRF jsou také vybaveny např. rozšířenými paměťmi, vstupy a výstupy. Nejčastější požadovanou funkcí brány je tzv. datalogger, který shromažďuje události a data přes IQRF Cloud.



### 3.2 IQRF IDE

Společnost IQRF Tech s.r.o. poskytuje ke všem svým produktům integrované vývojové a servisní prostředí tzv. IDE (Integrated development and service environment). Součástí IDE je i grafické uživatelské prostředí – GUI. Tento softwarový nástroj je kompatibilní s Windows 10, Windows 8.x, Windows 7 a Windows Vista. Pro jeho instalaci je potřeba driverů pro USB a kompilátor jazyka C.

IDE slouží jako prostředí pro tvorbu aplikací pro IQRF a zahrnuje všechny nástroje pro jejich vývoj. IDE má úlohu programátoru přijímače/vysílače a debuggeru. Dále uživateli poskytuje RF terminál pro ovládání sériové komunikace, správce IQMESH sítě – nástroj pro správu, vizualizaci a testování DPA protokolu, dále má IDE funkci servisního nástroj IQRF, který zahrnuje konfigurator, analyzátor, tester a skener. Součástí tohoto prostředí je i nástroj pro aktualizaci operačního systému přijímačů/vysílačů, firmwaru a dalších pomocných nástrojů.



Obrázek 9: Ukázka GUI od IQRF IDE

## 4 Rodina mikrokontrolérů NXP Kinetis L

Původním výrobcem produktů řady Kinetis L je společnost Freescale. Avšak v roce 2015-2016 došlo k fúzi společností Freescale a NXP, a proto lze nyní vidat tyto produkty původně od Freescale pod značkou NXP.

Rodina společnosti NXP Kinetis L zahrnuje více než 200 vysoce výkonných mikrokontrolérů. Celkově je tato rodina rozdělena do šesti kategorií. MCU jsou vystavěny na 32bitové architektuře s procesorovým jádrem ARM Cortex-M0+. Řada MCU Kinetis L se prezentuje nízkou spotřebou, ale i přesto tyto procesory disponují značným výkonem. Mezi standardní vybavení této řady patří nízko odběrové sériové asynchronní rozhraní UART a časovače, PWM, sériové periferní rozhraní SPI, sběrnice I2C, hodiny reálného času RTC (Real-time clock) a analogový komparátor.

Jednotlivé kategorie MCU této rodiny se liší například ve velikosti taktovací frekvence jádra procesoru, ve velikosti paměti Flash a SRAM, vybavenosti periferiemi a v počtu pinů. Podrobnější popis s dalšími parametry je uveden na obrázku 10 a níže v textu.

Products	CPU	Memory	Packages		Comms		HMI		Security	
	Arm Cortex-M0+		Type	Pin Count	I <sup>2</sup> S	USB Full-Speed	FlexIO*	Segment LCD	HW Encryption	Tamper Detection
<b>KL8x</b> > Security	72 MHz (Up to 96 MHz)	128 KB Flash 96 KB SRAM	LQFP, MAPBGA, WLCSP	64–121	✓	✓	✓	✓	✓	✓
<b>KL4x</b> > USB and Segment LCD	48 MHz	128–256 KB Flash 16–32 KB SRAM	LQFP, MAPBGA	64–121	✓	✓	✓	✓		
<b>KL3x</b> > Segment LCD	48 MHz	32–256 KB Flash 4–32 KB SRAM	LQFP, MAPBGA	64–121	✓		✓	✓		
<b>KL2x</b> > USB	48 MHz / 72MHz (Up to 96)	32–512 KB Flash 4–128 KB SRAM	LQFP, QFN, MAPBGA, XFBGA, WLCSP	32–121	✓	✓	✓		✓	
<b>KL1x</b> > Mainstream	48 MHz	32–256 KB Flash 4–32 KB SRAM	LQFP, QFN, MAPBGA, XFBGA, WLCSP	32–80	✓		✓			
<b>KL0x</b> > Entry-level	48 MHz	8–32 KB Flash 1–4 KB SRAM	LQFP, QFN WLCSP	16–48						

Obrázek 10: Přehled podskupin rodiny Kinetis®[14]

## 4.1 Kategorie v rodině Kinetis L

Podstatou řady je sdílení některých svých rysů, které jsou popsány níže v textu. Jednotlivé kategorie se pak liší svým účelem použití, výkonem, který se zvyšuje s rostoucím číslem kategorie. Příslušníci daných kategorií jsou opět rozlišeni číslováním, které se stupňuje s jejich parametry. Společné rysy Kinetis L:

- Procesor s jádrem Cortex-M0+
- Takt procesoru 45 MHz
- Několik režimů úspory energie LLWU
- Odpojitelnost periférií od zdroje hodinového signálu tzv. *clock gating*
- Napěťová logika 3,3 V
- Non-volatilní paměť Flash technologie TFS (*thin film storage*) 90 nm
- Operační paměť statického typu
- Ochranné principy v paměti
- Přímý přístup do paměti z periférií DMA (*direct memory access*)
- A/D převodník s šířkou slova 12 nebo 16 bitů
- D/A převodník s šířkou slova 12 bitů
- Rychlý komparátor
- Asynchronní sběrnice UART
- Synchronní sběrnice I2C a SPI
- Hodiny reálného času
- Nízkoodběrové časovače, včetně 16-bitových TPM (*timer pulse-width modulator*) modulů
- Časovač periodického přerušení s limitací 32 bitů

#### **4.1.1 KL0x**

V rámci řady Kinetis L se jedná o nejnižší kategorii, což se projevuje i na její ceně, která je její značnou dominantou. Tato kategorie poskytuje zákazníkům jednodušší přechod z 8bitových mikroprocesorů, které mají stejné rozmístění vývodů. S tím je spojena velikost pouzdra čipu (4x4 mm), které zůstává rovněž neměnné. Výjimku této kategorie tvoří její nejnižší produkt, který nedisponuje třemi, jindy společnými, rysy. Chybějící části je číslicově-analogový převodník, Direct Memory Access (DMA) a Low Leakage Wakeup Unit (LLWU).

#### **4.1.2 KL1x**

Tato vyšší verze disponuje poměrně výraznými rozdíly od verze předchozí, a to nejen pro větší počet vývodů, ale také díky zvýšení velikosti programové a operační paměti. Změnou u některých, ne však u všech, produktů této řady je také nahrazení 12bitového AČ převodníku 16bitovým.

#### **4.1.3 KL2x**

Do této kategorie spadá i mikroprocesor, který je použitý v této diplomové práci. Zásadní změnou, ve srovnání s nižší kategorií, je použití normativní specifikace USB On-The-Go (neboli USB OTG či jen OTG). Díky této specifikaci je dovoleno připojit k MCU různá zařízení.

#### **4.1.4 KL3x**

Značnou výhodou této rodiny oproti předchozím kategoriím je přidání segmentového řadiče LCD displeje, který může disponovat až 375 segmenty. Displej je pak umístěn přímo na desku plošného spoje MCU.

#### **4.1.5 KL4x**

Nejvyšší kategorie v řadě Kinetis L kombinuje velmi nízký odběr s bohatou sadou analogových, komunikačních, časovacích a řídicích periférií jako je například LCD řadič a USB OTG.

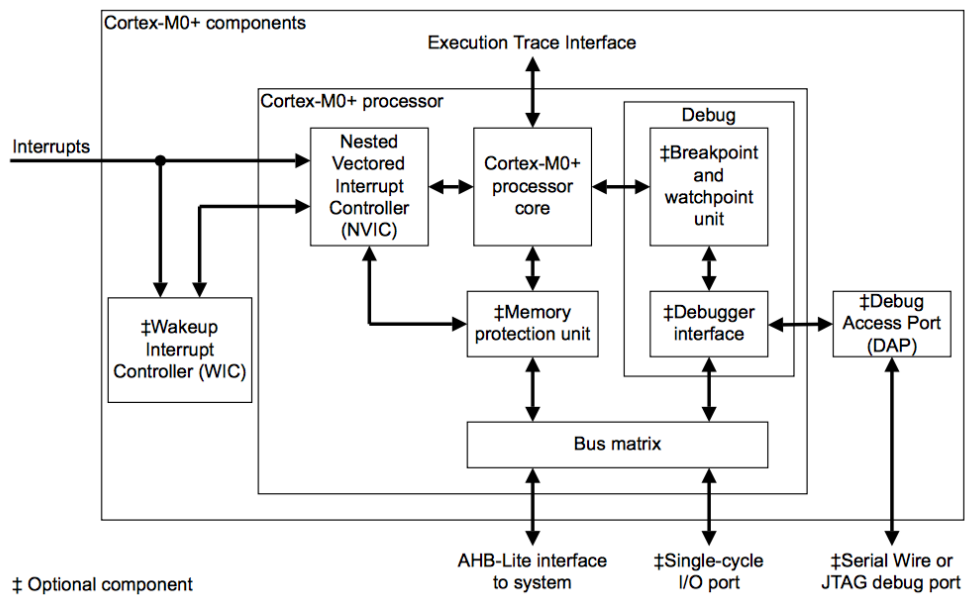
#### **4.1.6 KL8x**

KL8x je aktuálně nejvyšší kategorií rodiny Kinetis L a vyznačuje se nejen velmi nízkou spotřebou energie, ale i mnohými bezpečnostními funkcemi. Mezi tyto funkce patří detekce poškození, skutečný generátor náhodných čísel a nízko odběrové šifrovací jádro.

### **4.2 Jádro ARM Cortex-M0+**

Jádro procesoru ARM Cortex-M0+, dále zmiňováno už pouze jako procesor, vychází z úspěšné předchozí řady procesorů Arm Cortex-M0. Nadále je součástí tohoto konfigurovatelného procesoru

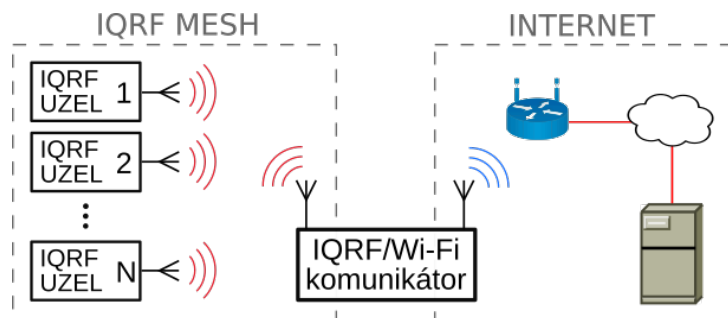
redukovaná instrukční sada tzv. RISC. Vzhledem k vysoké energetické účinnosti jej lze využít v mnoha aplikacích jako jsou MEMS senzory, Power management, zdravotnické a osobní monitory a také v neposlední řadě právě v nízko odběrových MCU.



Obrázek 11: Vnitřní struktura jádra ARM Cortex-M0+[4]

## 5 Návrh a realizace hardwaru IQRF - Wi-Fi komunikátoru

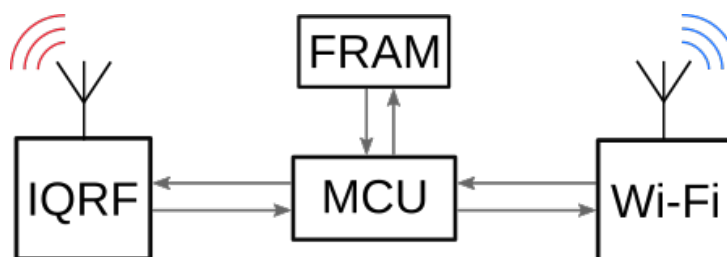
Komunikátorem v rozsahu této práce je myšleno vytvořené zařízení, jež z IQRF MESH sítě je schopno sbírat hodnoty z měřicích uzlů a skrze bezdrátové Wi-Fi připojení je posílat na internetové datové úložiště. Schématické znázornění topologie použití je vyobrazeno na obrázku 12. Před samotným návrhem komunikátoru bylo potřeba vzít v úvahu jednak umístění v rámci síťové topologie, tak i požadavky, které jsou na takové zařízení kladeny. Jedním z základních požadavků je robustnost a energetická nenáročnost.



Obrázek 12: Topologie použití komunikátoru

### 5.1 Návrh komunikátoru

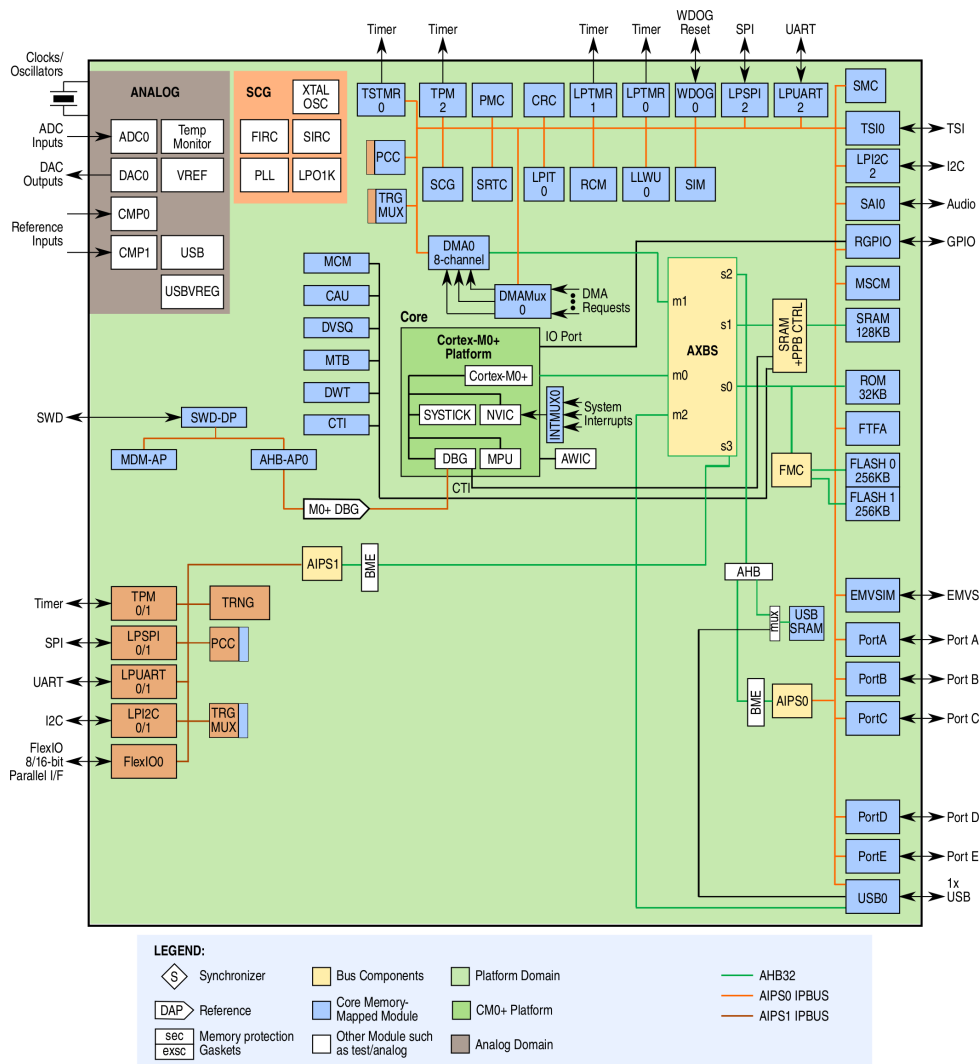
Jak již bylo zmíněno v kapitole zabývající se popisem technologie IQRF, pomineme-li P2P bezdrátové spojení, tak každá z IQRF sítě vždy obsahuje právě jeden uzel v roli koordinátora a až 239 koncových zařízení. Pro komunikátor se tedy jako výhodnější jeví role koordinátora, ke kterému se budou připojovat jednotlivé měřicí uzly a od nichž se budou sbírat data. Po IQRF modulu, druhou hlavní hardwarovou komponentou je mikrokontrolér, který bude zodpovědný jak za sběr dat, tak jejich následné odeslání. Za účelem připojení k internetu byl zvolen Wi-Fi modul. Dočasné odkládání dat a uložení konfigurace umožní energeticky nezávislá paměť. Z důvodů nízké spotřeby, na kterou je tato práce zaměřena, byla vybrána feroelektrická paměť.



Obrázek 13: Blokové schéma hlavních částí komunikátoru

### 5.1.1 Mikrokontrolér KL28Z512VLL7

Jedním z ústředních prvků komunikátoru je Mikrokontrolér. Ze zadání plyne, že se jedná o produkt firmy NXP z rodiny Kinetis L. Konkrétně vybraným modelem je MKL28Z512VLL7. Stejně jako celá řada L, tento MCU obsahuje procesorové 32bitové jádro ARM Cortex M0+. To je určeno především pro nízkoodběrový aplikace s čímž souvisí několik režimů, ve kterých může běžet. Samotné jádro obsahuje nástroje přímo určené pro běh operačního systému, jako například časovač SYSTICK, zpravidla používaný pro generování časové základny kernelu OS.



Obrázek 14: Blokový diagram struktury KL28Z512VLL7[4]

Mikrokontrolér MKL28Z512VLL7 je vybaven 512 kB programové flash paměti, 128 kB operační paměti SRAM a nemalým množstvím periférií. Ze sběrnic obsahuje 3xLPUART, 3x LPSPI, 3xLPSPI, 1xI2S, 1xUSB. Krom zmíněného SYSTICKu, možnosti pro hardwarové časování rozšiřuje 3xTPM a 2xLPTMR. V případě potřeby počítat cyklické redundantní součty je MCU vybaven vyhrazenou CRC periférií, která tyto výpočty provádí bez vytěžování CPU. Přímý

přístup periférii bez účasti procesorového jádra umožňuje 8kanálové DMA. Vzhledem ke skutečnosti, že jádro ARM Cortex M0+, respektive jeho řadič přerušení NVIC má pouze 32 vstupů, byl do MCU zabudován multiplexor přerušení INTMUX. Ten umožňuje vybavení mikrokontroléru, takovým množstvím periférií.[4]

### 5.1.2 Wi-Fi modul RN-131C

Připojení komunikátoru k internetu, potažmo k vzdálenému datovému úložišti je realizováno pomocí Wi-Fi modulu. Byl vybrán model RN-131C od firmy Microchip. Přestože umožňuje samostatné fungování bez nadřazeného systému, v případě navrženého komunikátoru je použit jen jako prostředek pro přístup na vzdálený internetový server. Vybraný modul je certifikován pro provoz pod standardy IEEE 802.11b/g (oba 2,4 GHz), a to jak v infrastrukturním, tak ad-hoc režimu. Mezi podporované druhy zabezpečení bezdrátových sítí patří WEP-128, WPA-PSK (TKIP) i WPA2-PSK (AES). Samozřejmostí je pak vestavěná podpora režijních protokolů DHCP, DNS a ARP.



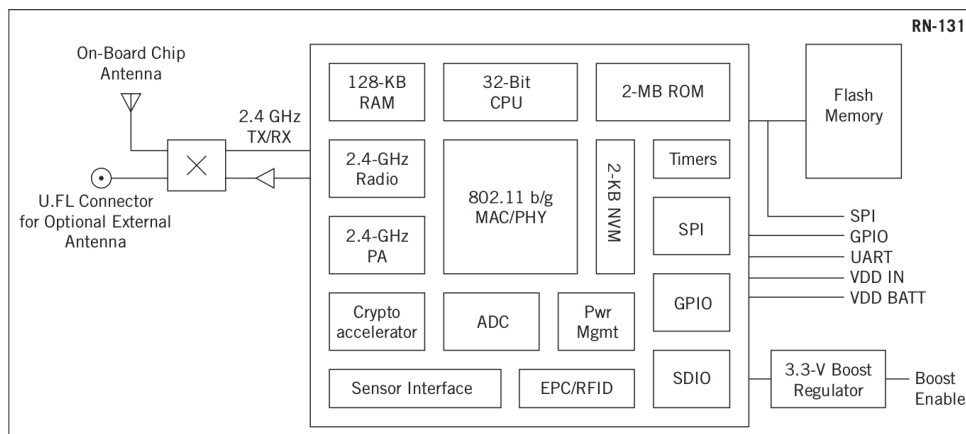
Obrázek 15: Vybraný Wi-Fi modul RN-131 [10]

Napájení je možné realizovat stabilizovanou hladinou 3,3 V, ale také prostřednictvím akumulátoru (2 až 3 V). Modul pro tyto potřeby obsahuje DC-DC měnič, jehož výstupní napětí 3,3 V je možné využít i pro ostatní obvody.

Výrobce uvádí vhodnost použití zejména v oblastech vzdáleného monitorování, telemetrii, průmyslových senzorů a zdravotnických zařízení. Díky integrovaným hodinám reálného času RTC, nabízí Wi-Fi modul možnost usnutí do režimu velmi nízké spotřeby (4  $\mu$ A).

Mezi další funkcionality, jimiž RN-131 disponuje je 8 senzorických analogových vstupů, 10 vstupně-výstupních GPIO a sériová sběrnice SPI. Pro nenáročné senzorické aplikace lze navíc využít možnosti nahrání uživatelského firmwaru, čímž se eliminuje nutnost přidání řídicího mikrokontroléru. V rámci návrhu komunikátoru byla využita druhá z nabízených komunikačních rozhraní, a to sice UART. Konfigurace je možná dvěma způsoby. První z nabízených možností je využít sériový terminál pro komunikaci skrze UART. Druhou cestou je síťové připojení přes telnet. V obou případech se nastavení provádí pomocí jednoduchých textových příkazů.[10]





Obrázek 16: Blokový diagram Wi-Fi modulu RN-131 [10]

### 5.1.3 IQRF modul DCTR-76D

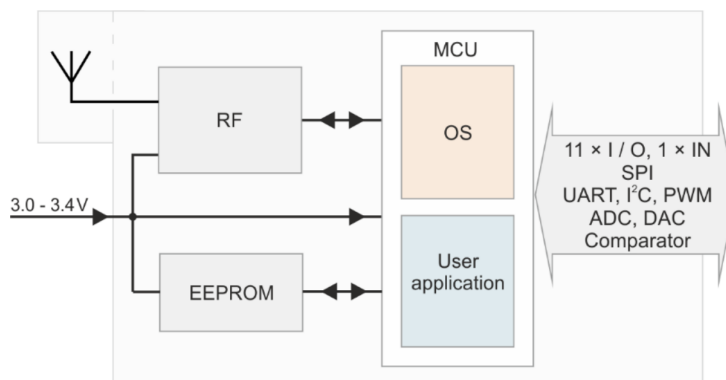
Spojení komunikátoru s IQRF MESH sítí je realizováno pomocí IQRF modulu DCTR-76D. V současné době firma Microrisc doporučuje pro nově navrhované zařízení jeden ze čtveřice nabízených modulu DCTR-77D, DCTR-76D, DCTR-75D, DCTR72D. Srovnání zmíněných možností je uvedeno v tabulce 2. Jelikož předpokladem je, že komunikátor bude uzavřené zařízení ubírala se volba směrem k modelům, které umožňují SMT montáž na DPS. Tento požadavek splňují pouze dva moduly. Jelikož dosah Wi-Fi sítí není velký a zpravidla se vyskytují v blízkosti zástavby, bylo vhodné upřednostnit IQRF modul s větším vyzařovacím výkonem, a proto byl vybrán DCTR-76D.

Tabulka 2: Srovnání aktuálně doporučovaných IQRF modulů

	DCTR-77D	DCTR-76D	DCTR-75D	DCTR-72D
Počet I/O	12	12	10	4
Vyzař. výkon	4 mW	10 mW	10 mW	10 mW
Frek. pásmo	868 MHz	433/868/916 MHz	868/916 MHz	433/868/916 MHz
Použitý MCU	PIC16LF1938	PIC16LF1938	PIC16LF1938	PIC16LF938
SMT montáž	ano	ano	ne	ne
Napět. reg.	ne	ne	ne	ano

DCTR-76Dx je vyráběn ve 4 variantách. První z nich (použita v této práci) neobsahuje anténu, RF signál má vyveden na pájecí plošku modulu a je určen pro frekvenční pásma 868/916 MHz. Mezi těmito pásmy lze modul softwarově přepínat. Druhá produkovaná varianta s označením "A" je totožná s předchozí, pouze s výjimkou antény, která je vytvořena na DPS jako součást modulu. Třetí varianta s označením "433" nemá vestavěnou anténu, avšak pracuje ve frekvenčním pásmu 433 MHz. Poslední prodávanou verzí je "IL" varianta. Ta je určený pro izraelský trh, kde jsou určitá omezení v používání frekvenčního pásma 916 MHz.

Kromě pro IQRF typických komunikačních rozhraní UART a SPI, je vybraný modul vybaven rovněž I<sup>2</sup>C. Mezi další funkcionality patří hardwarový časovač, AČ a ČA převodník.



Obrázek 17: Blokový diagram DCTR-76D[15]

## 5.2 Realizace HW komunikátoru

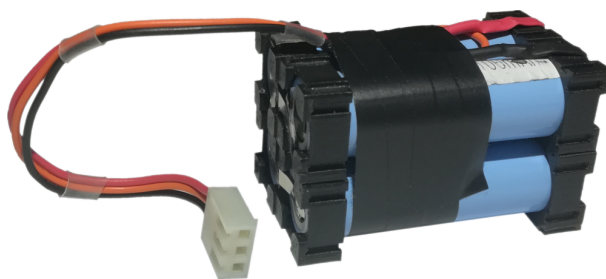
Pro zajištění nízké spotřeby a kompaktního řešení, je komunikátor realizován jako deska plošných spojů, kterou lze následně umístit do vhodné krabice pro ochranu před povětrnostními a mechanickými vlivy. Zapojení veškerých integrovaných obvodů vychází z doporučení, které jsou uvedeny v dokumentaci výrobce.

### 5.2.1 Napájení

Aby bylo zajištěno všestrannější použití komunikátoru s možností situování ve venkovních prostorech, je zařízení vybaveno vlastním elektrochemickým zdrojem energie. V dnešní době je na trhu nabídka nepřehledného množství typů akumulátorů, jež se od sebe liší jednak použitou chemií, tak i prostorovým uspořádáním elektrod a elektrolytu. Mezi běžně používané typy patří olovené (Pb), nikl-kadmiové (NiCd) nebo nikl-metal-hydridové (NiMH). Ovšem tyto druhy akumulátorů jsou spíše na ústupu, jelikož je postupně napříč celým spektrem aplikací nahrazují akumulátory na bázi lithia. Odměnou je především daleko vyšší energetická hustota, avšak přináší s sebou i určitá omezení a nároky na dodatečnou elektroniku nutnou pro provoz. Tyto obvody musí zahrnovat přepětovou, podpětovou, nadproudovou a teplotní ochranu. Přestože vyšší energetická hustota je žádoucí vlastnost z pohledu použití (jelikož zabírá méně prostoru), z bezpečnostního hlediska je to spíše přítěž. Při mechanickém poškození, nadměrném přepětovém, nadproudovém nebo teplotním zatížení dochází k silné exotermní reakci, která ve formě tepla uvolní energii uloženou uvnitř článku. Z těchto důvodů je nutno tyto veličiny měřit a při překročení mezních hodnot musí dojít k elektrickému odpojení akumulátoru. Pro zajištění udávané životnosti je též nutné zabezpečit hlídání napětí u vybíjení. Při větším poklesu pod minimální napětovou hladinu dochází k nevratným chemickým změnám, které mají za následek ztrátu

kapacity a nárůst vnitřního odporu, jinak typické příznaky pro stárnutí. Mezi běžně používané chemické principy patří lithium-iontová (Li-ion), lithium-polymerová (Li-pol) a lithium-železo-fosfátová (LiFePO<sub>4</sub>) technologie.

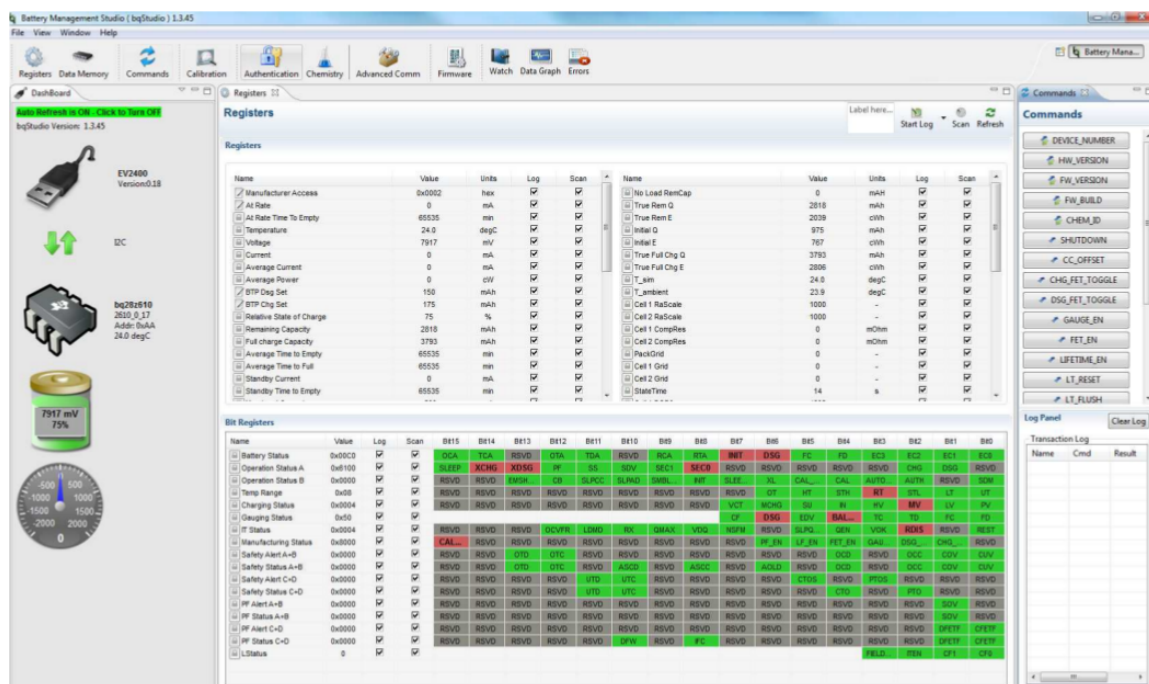
Pro navrhovaný komunikátor byly vybrány články Li-ion LG INR18650 MH1. Jedná se o typ s válcovitou konstrukcí a jmenovitou kapacitou 3200 mAh a napětím 3,67 V. Z katalogového listu prodejce GWL Power, a.s. lze dále vyčíst hodnotu minimálního napětí (tzv. "cut off") rovnou 2,5 V, maximální napětí při nabíjení 4,2 V, typický nabíjecí proud 1,55 A (maximálně 3,1 A) a typický vybíjecí proud 0,62 A (maximálně 10 A). Společným rysem všech vybraných a dále popisovaných komponent je možnost napájení 3,3V hladinou, čehož je v této práci využito a návrh napájení byl směřován touto cestou. Provozní napěťový rozsah vybraného článku je tedy od 2,5 V do 4,2 V. Takový rozptyl pro samotné integrované obvody není vhodný a bylo nutné hledat řešení, jak zajistit konstantní hladinu napájecího napětí. Jednou z možností bylo použití v sérii jen jednoho článku a do komunikátoru integrovat tzv. "buck-boost" DC-DC měnič, který dokáže kompenzovat variabilitu v určitém napěťovém okně okolo pracovního požadovaného napětí. Pro účely návrhu a realizace komunikátoru bylo ale zvoleno řešení s dvěma sériovými řazenými články, díky čemuž se napěťové okno akumulátoru zvýšilo na 5 V až 8,2 V. Následně stačilo zakomponovat do napájecí části popisovaného zařízení napěťový stabilizátor. Zvýšení kapacity je zajištěno paralelním přidáním dalších dvou článků. Výsledná konfigurace 2P2S nabízí jmenovitou kapacitu 6,4 Ah. Se sériovým řazením více článků uvnitř akumulátoru však vyvstávají další požadavky. Mimo měření napětí na každém článku zvlášť, také problém balancování. Jelikož při výrobě je nemožné zajistit naprosto stejné parametry produkovaného zboží, každý článek má lehce odlišnou kapacitu a samovybíjení. Nabíjení i vybíjení baterie způsobuje, že do/z každého sériově řazeného prvku je dodáván/odebírán totožný proud, ovšem různé kapacity neodvratně vedou k odlišnému napětí na jednotlivých článcích. Běžně vyráběné IO pro monitorování lithiových baterií tuto funkcionalitu nabízejí. Typicky při nabíjení, článek s nejnižší kapacitou dosáhne maximálního napětí jako první, přičemž ostatní nemusí být zdaleka nabitý na maximální úroveň. Tato situace je řešena paralelním připojením balančního rezistoru k pólům článku, tak aby se vybil na nejnižší napěťovou úroveň zapojenou v sérii.



Obrázek 18: Vytvořený akumulátor pro napájení komunikátoru

Zajištění výše popsaných ochranných opatření je realizováno pomocí integrovaného obvodu

BQ28Z610 produkovaného firmou Texas Instruments. Jedná se o autonomní obvod zajišťující monitorování a ochranu Li-ion baterie s jedním nebo dvěma sériově řazenými články. Univerzálnost použití pro různé konfigurace a výrobce je dosažena možností změny parametrů. Komunikační rozhraní I2C s frekvencí do 400 kHz, jednak umožňuje vyčítání měřených a spočtených hodnot, ale i kompletní konfiguraci. Pomocí servisního konfiguračního rozhraní EV2400 a softwaru Battery Management Studio, lze do integrovaného obvodu nahrát chování konkrétní chemie. Je to nezbytné pro správné fungování odhadovacích algoritmů pro stupeň nabití SOC. Další možnosti nastavení tvoří limity pro COV, CUV, COC, balancování, povolení alarmů a také kalibrace. Schéma zapojení je zobrazeno na obrázku 20. Vychází z doporučení uvedených v katalogových listech pro BQ28Z610 a také ze zapojení vývojové desky BQ28Z610EVM.

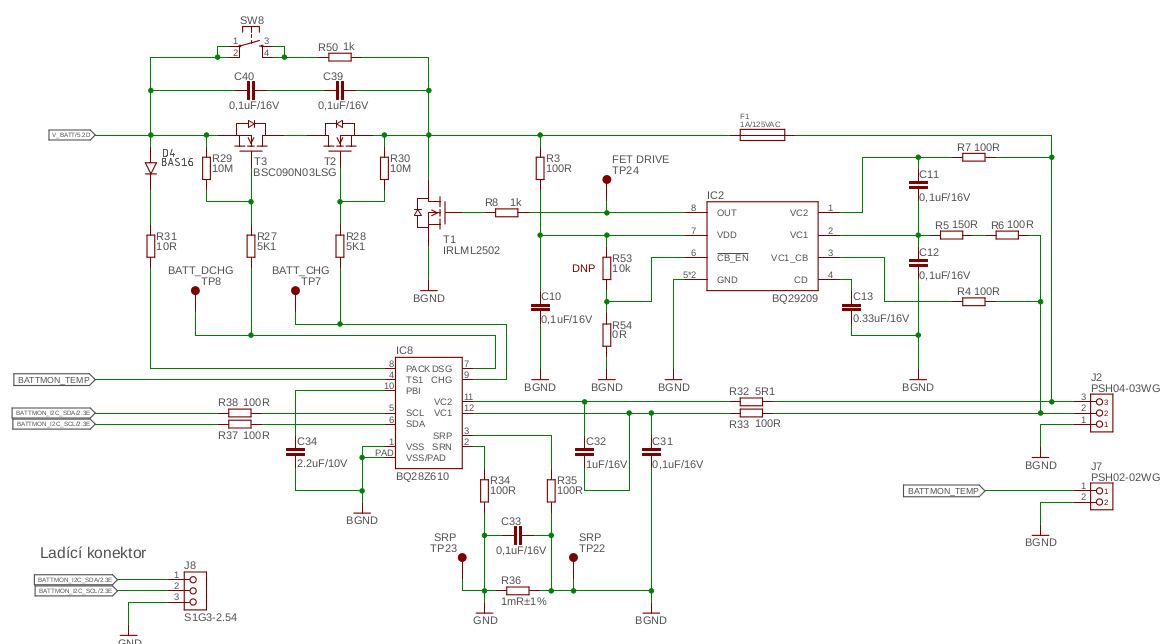


Obrázek 19: Snímek obrazovky Battery Management Studia

Napájení samotného IO probíhám přímo z akumulátoru. Dvojice unipolárních N-kanálových MOSFET tranzistorů T2 a T3 v antisériovém zapojení tvoří hlavní spínací prvky pro napájení ostatních součástí na DPS komunikátoru. Toto zapojení je velmi výhodné, jelikož zde separátně odpojovat nabíjení a vybíjení akumulátoru. Otevření tranzistoru T3 umožní proudit též z článků skrz diodu uvnitř tranzistoru T2, opačným směrem však nikoliv. Při otevření T2 a zavření T3 je situace opačná. Za běžného provozu, kdy všechny články jsou uvnitř svého provozního okna jsou průchozí oba tranzistory, čímž se minimalizují ztráty na spínacích prvcích. Vybrané MOSFETy BSC090N03LSG mají při svém plném otevření mezi elektrodami S ("source") a D ("drain") odpor pouhých 7,5 mΩ. Ovšem zprůchodnění indukovaného kanálu vyžaduje přivedení kladné hodnoty napětí mezi G ("gate") a D. To vyžaduje vyšší napětí, než které je na akumulátoru. Jeho vytvoření je zajištěno integrovanou nábojovou pumpou uvnitř BQ28Z610. Měření proudu

je realizováno precizním nízkoodporovým rezistorem R36 (tzv. "SHUNT") s hodnotou odporu 1 m $\Omega$  a tolerancí pouhé 1 %.

Pro potřeby balancování je integrovaný obvod BQ28Z610 vybaven vnitřními tranzistory, které při aktivaci připojí paralelně k článku balanční rezistor R33 (tzv. "bypass"). V případě poklesu napětí na článku pod minimální hodnotu, dochází k uspání IO tak, aby se minimalizovala spotřeba energie z už tak vybitého akumulátoru. Zpětné probuzení nastane, při přivedení napětí za spínací tranzistory. To vytváří problém při vyjmutí akumulátoru a jeho opětovném připojení. I když je nabitý, pro probuzení je nutné vytvořit napěťový impulz na PACK vstupu. K tomu slouží tlačítko SW8. Za účelem ladění a konfigurace je DPS komunikátoru vybavena ladícím konektorem J8, přes který je možné připojit EV2400.

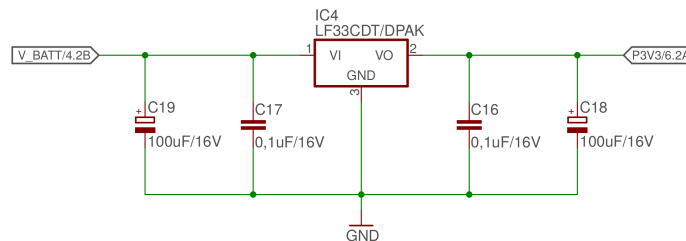


Obrázek 20: Schéma zapojení integrovaných obvodů pro ochranu akumulátoru

V katalogovém listu se dále uvádí, že je doporučeno zařízení vybavit ochranou druhého stupně. Ta spočívá v zakomponování tavné proudové pojistky, která při selhání prvotní ochrany, ve smyslu samočinného odpojení z důvodu přepětí, pojistku přepálí a tím trvale akumulátor odpojí. Výrobce doporučuje svou řadu integrovaných obvodů BQ2920X. Jelikož tato ochrana je zamýšlena jako poslední instance před potenciálním vznícením akumulátoru, je vyhodnocování nadlimitních podmínek prováděno čistě hardwarově bez jakýchkoliv možností nastavení nebo změny konfigurace. To však vzhledem k tomu, že Li-ion články různých výrobců mohou mít různé limitní hodnoty znamená, že tyto integrované obvody nejsou univerzální. Vybraným INR18650 MH1 vyhovuje BQ29209 s prahovým napětím 4,3 V. Změnou kapacity kondenzátoru C13 lze upravovat dobu, za kterou dojde k aktivaci výstupu při překročení prahového napětí. Zvolená hodnota 0,33  $\mu$ F odpovídá reakční době 3 s, po které je výstup OUT přepnut do logické 1. Díky tomu je otevřen

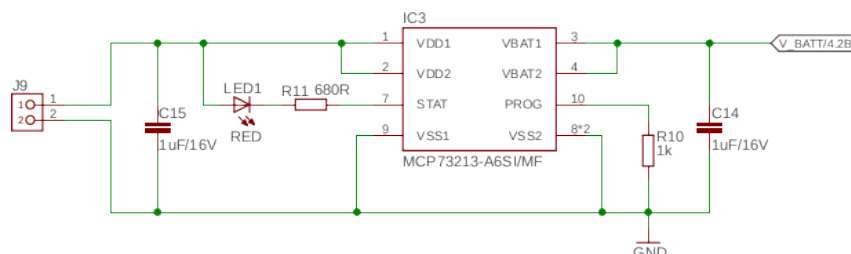
tranzistor T1, který zajistí zapojení akumulátoru nakrátko jen skrze tavnou pojistku, jež je následně přepálena. Další funkcionalitou, kterou disponuje BQ29209 je možnost balancování, avšak toto je již zajištěno pomocí IC8. Schéma je navrženo tak, že při neosazení R53, dojde k deaktivaci, zatímco při vynechání R54 lze funkcionalitu aktivovat pro případné testování.

Jak již bylo zmíněno v počátku této podkapitoly, pro snížení hodnoty napětí byl v případě této práce použit stabilizátor LF33CDT. Jedná se o lineární napěťový LDO regulátor vyráběný firmou STMicroelectronic. Pracovní okno pro vstupní napětí je od 4,3 V do 16 V, při maximálním výstupním proudu 1 A. Jak je patrné z obrázku 21 zapojení regulátoru je doplněno čtveřicí blokovacích a filtračních kondenzátorů, jež zabráňují rozkmitání regulačního obvodu uvnitř IO.



Obrázek 21: Schéma zapojení stabilizátoru na 3.3 V

V případě umístění ve venkovních prostorech je výhodné, aby DPS komunikátoru byla uzpůsobena k možnosti dobíjení akumulátoru bez nutnosti jeho vytažení. Pro zajištění dlouhodobého fungování se jako vhodné řešení jeví možnost dobíjení z obnovitelných zdrojů energie, jako je solární panel nebo malá větrná elektrárna. Z těchto důvodů byl do návrhu schématu zakomponován integrovaný obvod MCP73213 od firmy Microchip. Jedná se o lineární napěťový regulátor pro nabíjení Li-ionového dvoučlánekového akumulátoru. Zvolený IO je vybaven senzorem proudu a spolu se schopností měřit napětí, reguluje nabíjecí proud podle zabudovaného algoritmu. Změnou hodnoty rezistoru R10 lze měnit velikost maximálního nabíjecího proudu. Jelikož standardní nabíjecí proud vybraných článků je 1,55 A, je nabíječ nastaven na svou maximální proudovou hladinu 1,1 A. Jak je patrné na obrázku 22, MCP73213 ke své činnosti nepotřebuje již žádné další pasivní součásti kromě filtračních kondenzátorů na vstupu a výstupu. Pro indikaci nabíjení slouží LED dioda LED1.



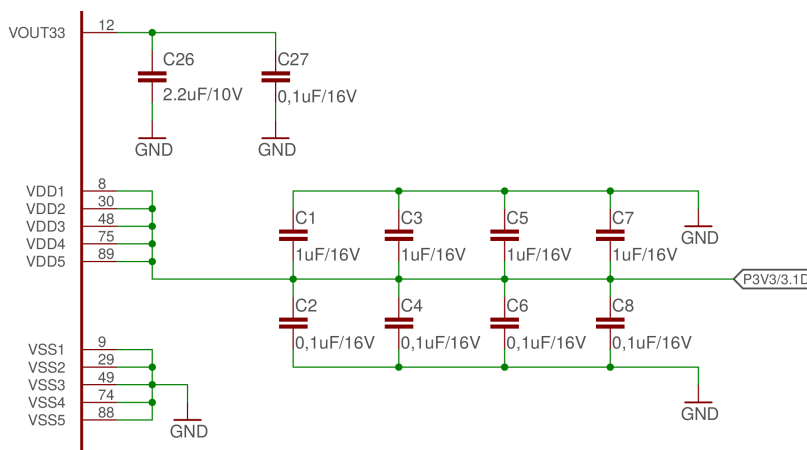
Obrázek 22: Schéma zapojení nabíječe akumulátoru

### 5.2.2 Mikrokontrolér

Elektronické zapojení bylo pro zajištění správní funkčnosti provedeno po vzoru vývojové desky FRDM-KL28Z.

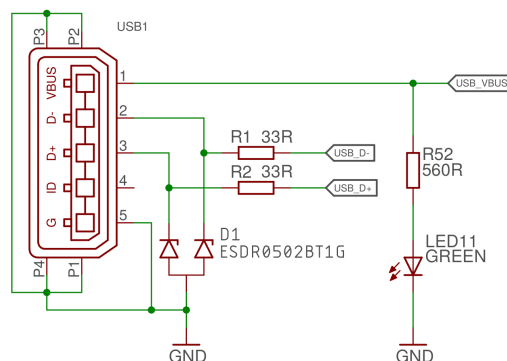
MCU má pro své napájení vyvedeno 5 dvojic napájecích pinů. To zajišťuje spolehlivý rozvod napětí napříč všemi periferiemi uvnitř pouzdra integrovaného obvodu. Na napájecí linky jsou co nejbližše pouzdru integrovaného obvodu připojeny blokovací kondenzátory o hodnotách  $1\ \mu\text{F}$  a  $0,1\ \mu\text{F}$ . Ty mají za cíl odstranit vysokofrekvenční rušení, které by mohlo nepříznivě ovlivnit ostatní obvody a zhoršit EMC výsledného zařízení. Jednou z funkcionalit vybraného mikrokontroléru je USB. Pro případ použití v aplikaci, kde by celé zařízení bylo napájeno z tohoto portu, je integrální součástí MCU napěťový regulátor na  $\approx 3,3\ \text{V}$ . Výstup napětí je z pouzdra vyveden pinem VOUT33 a vyžaduje připojení dvojice kondenzátorů C26 a C27.

Obrázek 23: Schéma zapojení napájecích pinů MCU



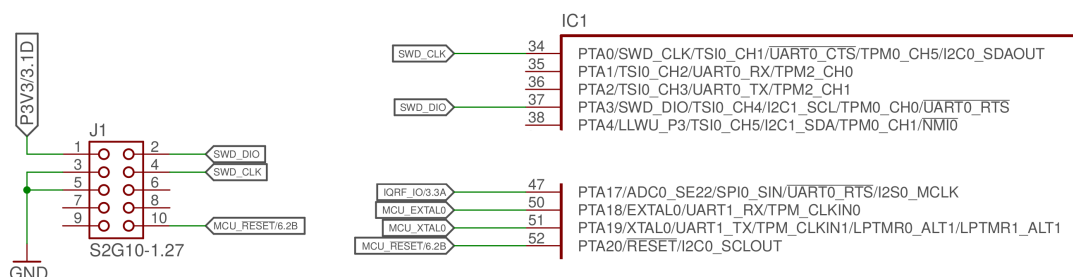
Pro připojení komunikátoru k počítači za účelem konfigurace slouží, nejběžnější používané, USB rozhraní. Konkrétně, na straně navrhovaného zařízení, se jedná o samici konektoru micro USB 2.0 typu B. Z 5 pinů, jenž se na konektoru nachází slouží krajní dva pro napájení hostovaného zařízení 5 V. Piny 2 a 3 jsou samotné datové vodiče pro sériovou komunikaci. Poslední vodič je v případě této práce nezapojen, jelikož se jedná o identifikátor pro OTG komunikaci. Ten komunikátor nepotřebuje, jelikož bude vždy vystupovat na sběrnici jako host. Pro ochranu MCU je mezi datové linky a zem připojena ESD ochrana v kombinaci se sériově zapojenými rezistory. Napájení z hostitelského zařízení je připojeno k VREGIN pinu na mikrokontroléru. Pro signalizaci řádně připojeného USB je komunikátor vybaven zelenou kontrolní LED diodou.





Obrázek 24: Schéma připojení USB rozhraní

Samotný mikrokontrolér KL28Z512VLL7 není vybaven, skrze USB, ladicím rozhraním openSDA. Pro tyto potřeby obsahuje vývojová deska FRDM-KL28Z dodatečný MCU MK20DX. Ten zprostředkovává připojení k ladicím funkcionalitám procesorového jádra skrze SWD. To slouží jako dvou vodičová alternativa k JTAG. Oba zmíněné standardy využívají stejný komunikační protokol. Pro potřeby brány je však zbytečné vybavovat DPS dalším mikrokontrolérem sloužícím jen pro účely ladění. Z těchto důvodů se na DPS nachází pouze konektor, jež vyvádí od CPU linky potřebné pro SWD.

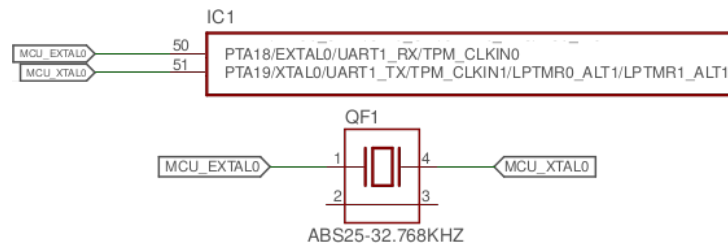


Obrázek 25: Schéma připojení SWD ladicího rozhraní

Hodinový signál pro všechny periferie, paměti a procesorové jádro mikrokontroléru zajišťuje SCG (system clock generator) modul. Ten obsahuje jednak 2 vnitřní nastavitelné RC oscilátory (vysokofrekvenční a nízkofrekvenční), vstup pro externí oscilátor SOSOC, soustavu děliček a násobičku PLL. Není-li vyžadováno velmi přesné časování pro jednotlivé periferie, vystačí integrované zdroje hodin. Pro taktování modulu hodin reálného času RTC bylo však vhodné připojit externí krystalový rezonátor s frekvencí 32,768 kHz, tak jako je zobrazeno na obrázku 26. Nastavením děličky uvnitř RTC na poměr 1:32768, je dosaženo frekvence 1 Hz, kterou je řízeno navýšování hodnoty ve vnitřním registru. Zatěžovací kondenzátory ke krystalu není nutné přidávat ve formě diskretních součástek, jelikož je možné pomocí registru nastavit kapacitu z kondenzátorové sítě



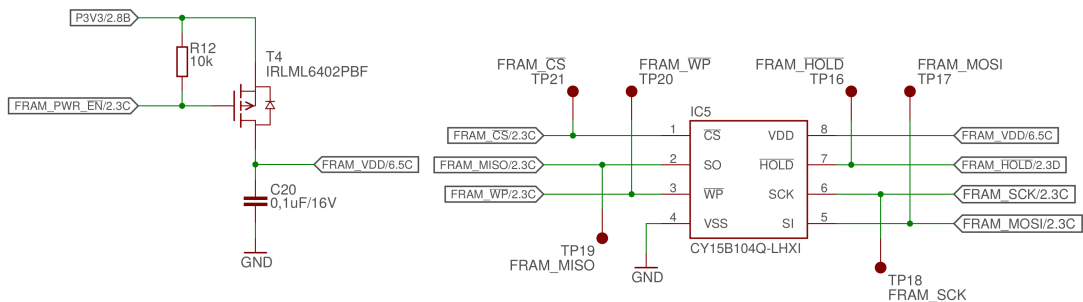
uvnitř integrovaného obvodu.



Obrázek 26: Schéma připojení rezonačního krystalu k MCU

### 5.2.3 FRAM paměť

Další z nezbytných komponent komunikátoru je energeticky nezávislá paměť. Ta slouží pro potřeby dočasněho ukládání naměřených hodnot z jednotlivých stanovišť a takové k uchování konfigurace. Byla vybrána feroelektrická paměť s kapacitou 4 MB od firmy Cypress. Jediné doporučení od výrobce, které se v dokumentaci uvádí, je umístění blokovacího kondenzátoru na napájecí linku do blízkosti pouzdra. Feroelektrické paměti se kromě velkého množství zapisovacích cyklů vyznačují také velmi nízkou spotřebou, která dle katalogového listu činí maximálně 3 mA. I přesto byl na napájecí linku (stejně jako v případě Wi-Fi) přidán P-kanalový MOSFET tranzistor, který řídí napájení. Komunikačním rozhraním pro přístup k paměti je v tomto případě SPI (MCU periferie LPSPI0).

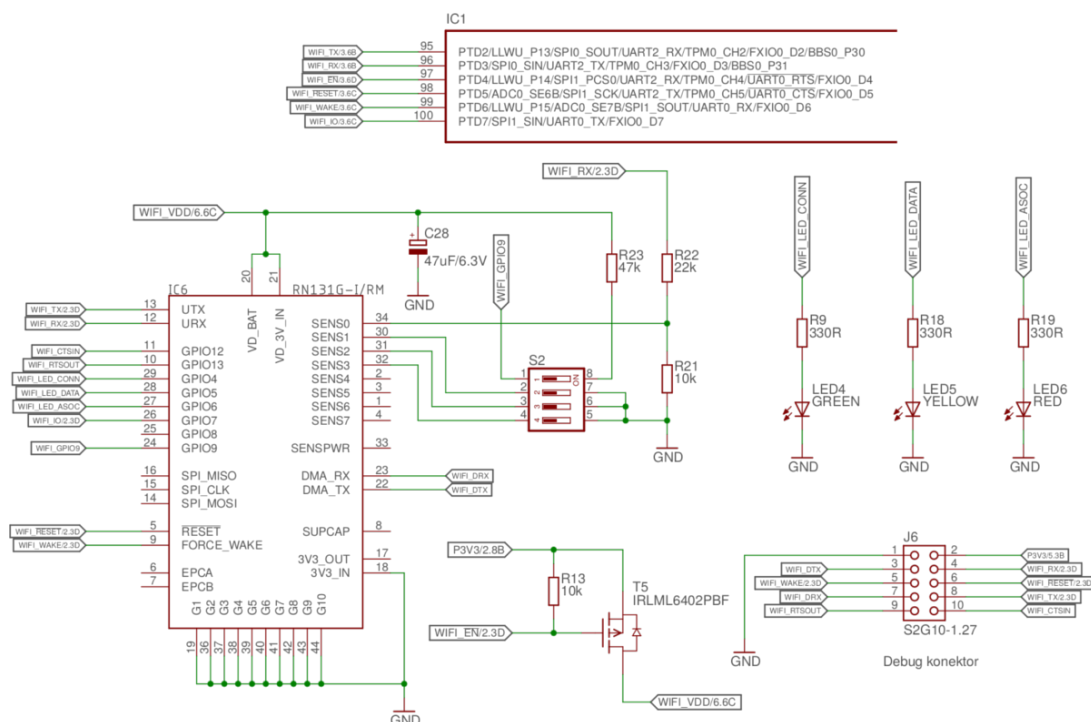


Obrázek 27: Schéma zapojení externí FRAM paměti CY15B104Q-LHX1

### 5.2.4 Wi-Fi modul

Schéma zapojení vychází z vývojové desky RN131/171 PICtail. Pro účely testování při vývoji firmwaru bylo zachováno připojení DIP přepínače S2 a ladícího konektoru J6. Pro komunikaci s mikrokontrolérem byla využita sběrnice UART, konkrétně LPUART2 periferie MCU. K indikaci stavu Wi-Fi modulu slouží trojice signalizační LED diod. Při rádiovém vysílání může spotřeba

stoupnout až na 212 mA ke kompenzaci těchto proudových špiček slouží elektrolytický kondenzátor C28 s kapacitou 47  $\mu$ F. Jelikož komunikátor byl navržen s ohledem na spotřebu elektrické energie, tak stejně jako u dalších komponent i napájení Wi-Fi modulu je řízeno unipolárním P-kanálovým MOSFET tranzistorem. Radiofrekvenční výstup není vyveden na DPS. RN-131 disponuje vlastní anténou a pro případ nutnosti připojit externí, je vybaven U.FI konektorem na své vrchní straně.

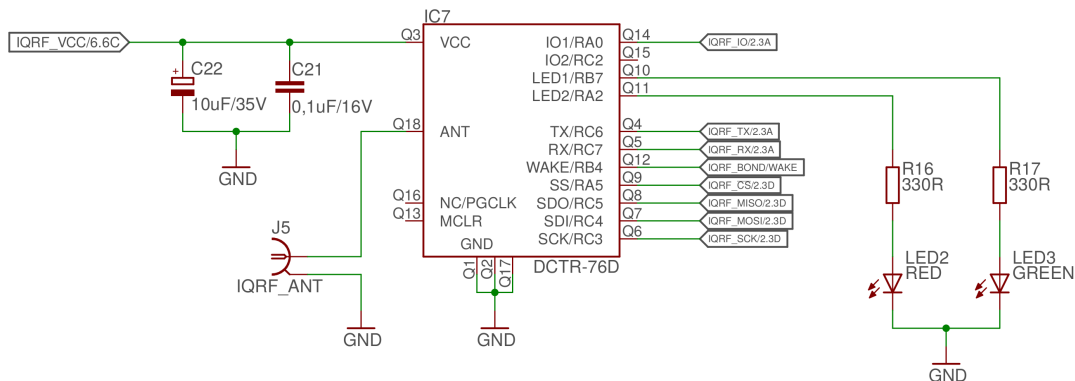


Obrázek 28: Schéma zapojení Wi-Fi modulu

### 5.2.5 IQRF modul

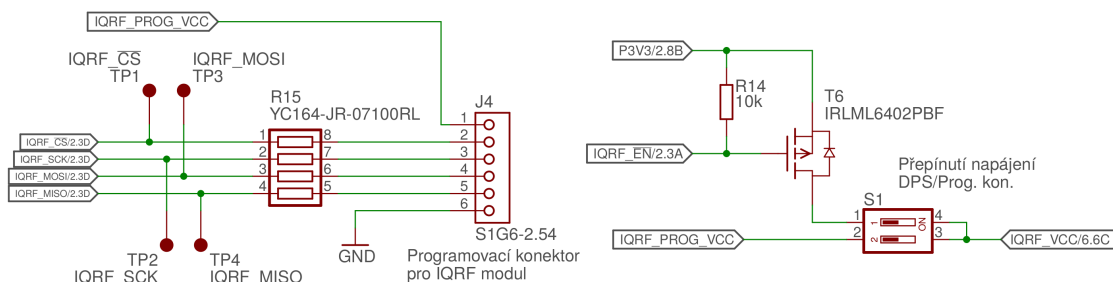
Vybraný IQRF modul DCTR-76D není vybaven vlastní anténou. Pin 19 je tedy připojen ke koaxiálnímu konektoru typu SMA, kde je přišroubována externí anténa. Stejně jako u všech ostatních integrovaných obvodů i zde je na napájecí lince blokovací kondenzátor, jež je navíc doplněn filtračním elektrolytickým kondenzátorem. Ten má za úkol filtrovat napětí při proudových špičkách způsobených vysíláním modulu. K signalizaci činnosti a stavu slouží dvojice LED diod (LED2 - červená, LED3 - zelená).

Spojení s MCU je realizováno pomocí sériových sběrnic SPI a UART (MCU periferie LPUART0 a LPSP11). Tyto komunikační standardy doplňuje signál pro bondování/probuzení z režimu spánku a jeden digitální vstupně/výstupní pin.



Obrázek 29: Schéma zapojení IQRF modulu DCTR-76D

Jelikož modul DCTR-76D se neusazuje do patice pro SIM kartu, tak jako některé jiné moduly, bylo potřeba najít jiný způsob, jak se připojit k modulu IQRF IDE skrze programátor. Pro tento účel byl navržen konektor J4. Na něj jsou vyvedeny všechny komunikační linky SPI spolu s napájením. Při připojení k programátoru je nutné, aby z něj byl IQRF modul i napájen. K tomu slouží DIP přepínač, pomocí něhož je možno odpojit napájení z DPS a připojit z programátoru. Stejně jako u FRAM paměti i tady v rámci minimalizace spotřeby je přívod řízen unipolárním tranzistorem, který je ovládán z MCU.



Obrázek 30: Schéma zapojení programovací konektoru a napájení pro IQRF modul

### 5.3 Ovládací a indikační prvky

Deska plošných spojů je vybavena celkem 5 tlačítky, 4 indikačními prvky v podobě LED diod a dvěma DIP přepínači.

Z uživatelského pohledu je nejvýznamnější tlačítko SW1, které zajišťuje Reset mikrokontroléru. Zbylá tlačítka SW4, SW5, SW6 a SW7 sloužila především pro účely testování během vývoje firmwaru. Nyní tato tlačítka nemají svou vlastní funkcionalitu, což však poskytuje jistou výhodu pro případné rozšiřování funkcí stávajícího zařízení. Každé tlačítko je vybaveno RC článkem pro odstranění zákmitů a spolu s pull-up rezistorem a bipolárním tranzistorem poskytují ochranu vstupů

mikrokontroléru např. proti elektrostatickým výbojům. Druhou možností, bez použití transilu, by bylo galvanické oddělení, které je však složitější (je potřeba větší množství součástek a s tím spojené vyšší náklady) a v této aplikaci je zbytečné. Tlačítko pro reset MCU je navíc vybaveno jednoduchým DIP přepínačem SW3, který slouží k permanentnímu podržení MCU v resetu, aby všechny IO porty MCU byly podrženy ve stavu vysoké impedance např. při připojování se k IQRF modulu skrze programovací konektor nebo k při připojování k Wi-Fi modulu skrze ladící konektor. V případě, že by toto bylo opomenuto, mohlo by dojít k nežádoucímu ovlivňování komunikace vlivem parazitního napájení mikrokontroléru z datových linek.

Další nezbytnou součástí je DIP přepínač S3, který umožňuje přepínání několika režimů komunikátoru, které budou popsány dále v textu. Funkce tohoto DIP přepínače spočívá v uzemňování interních pull-up rezistorů v mikrokontroléru, tedy uvedení signálu do stavu logické 0 při jeho sepnutí. Tento interní pull-up rezistor jinak udržuje na jednotlivých portech MCU logickou 1, a tím zabraňuje výskytu nedefinovaným stavům mezi logickou 0 a 1.

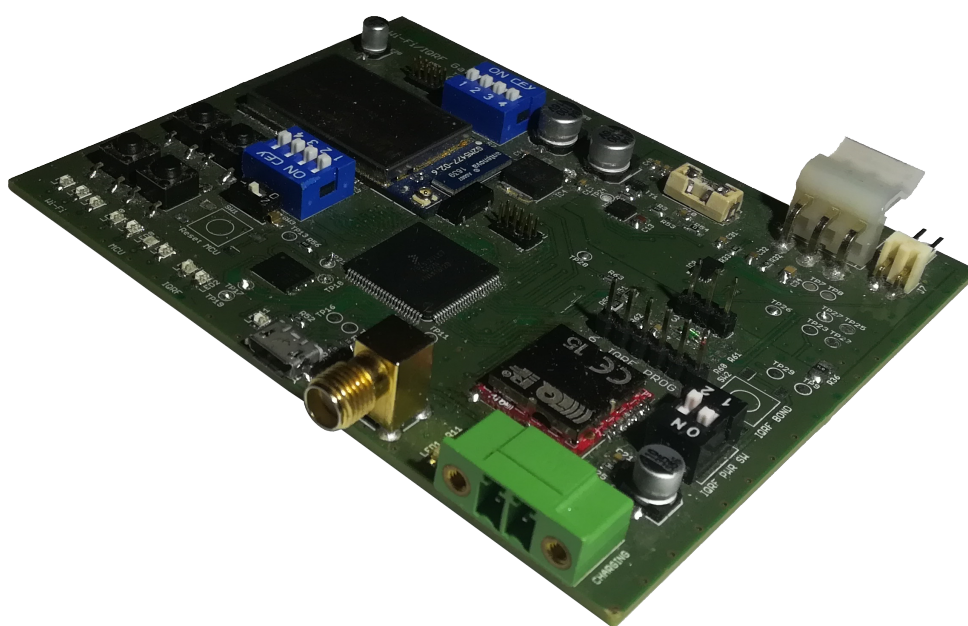
Indikační prvky tvoří dvě zelené diody LED8 a LED10, jedna červená dioda LED7 a jedna oranžová dioda LED9. Účelem těchto LED diod je signalizace jednotlivých režimů a případných chyb. Čtveřice LED diod je připojena ke GPIO mikrokontroléru přes rezistory, které omezuje protékající proud.

## 5.4 Zhotovený komunikátor

Deska plošných spojů komunikátoru je navržena pomocí editoru EAGLE V9.0.1 společnosti Autodesk®. Kompletní schéma použitých obvodů s vytvořeným návrhem DPS lze nalézt v příloze A.

Na základě GERBER dat, jež je možno exportovat ze softwaru EAGLE, byla českou společností PragoBoard s.r.o. vyrobena deska plošných spojů. Dvouvrstvá DPS je z obou stran pokryta měděnou fólií o tloušťce 17  $\mu\text{m}$  (tzv. jednouncovou mědí). Pro povrchovou úpravu DPS byla použita technologie žárovým pocínováním HASL (Hot Air Solder Leveling). Rozměry DPS činí 84x99 mm.

Upevnění baterií je zajištěno pomocí konstrukce vytisknuté na 3D tiskárně. Galvanicky poniklovaný kovový pásek pak zajistil vzájemné vodivé propojení článků. K těm byl přivařen pomocí odporové metody svařování.



Obrázek 31: Osazená DPS navrženého komunikátoru



Obrázek 32: Zhotová DPS uvnitř vybrané krabičky

## 6 Návrh a realizace firmwaru IQRF - Wi-Fi komunikátoru

Následující kapitola popisuje strukturu a realizaci firmwaru pro navržený hardware komunikátoru. Firma ARM Holdings, jakožto majitel duševního vlastnictví k procesorovým jádrům ARM<sup>®</sup> Cortex<sup>®</sup> M0+, dodává kompilátor spolu s linkerem. Jednotliví výrobci mikrokontrolérů a SoC (systém na čipu), tento kompilátor s linkerem přebírají a pouze jej integrují do svých vývojových prostředí, která jsou obvykle vytvořena na bázi Eclipse IDE. Takový koncept je s výhodou využíván zejména z důvodu otevřených zdrojových kódů Eclipse, který je tak snadno modifikovatelný pro potřeby výrobců. Ti pak pouze do softwaru integrují podporu a šablony projektů ke svým produktům spolu se správným nastavením kompilátoru a linkeru. NXP, jakožto výrobce použitého MCU, díky své fúzi s Freescale nabízí studia hned dvě - MCUXpresso a CodeWarrior<sup>®</sup>.

Vzhledem ke zvyklostem na katedře kybernetiky a biomedicínské inženýrství bylo pro účely vypracování této práce zvoleno Atollic TrueSTUDIO for ARM. Jednalo se o vhodnou volbu, jelikož toto IDE nabízí podporu pro široké portfolio produktů napříč všemi významnými výrobci. Avšak během vypracovávání této práce došlo k odkoupení zmíněného studia firmou STMicroelectronics, která pro nové verze odstranila podporu konkurenčních výrobců.

Prostřednictvím vedoucího práce byla pro zvolený mikrokontrolér poskytnuta šablona projektu, jejímž autorem je Ing. Martin Stankuš, Ph.D. Ta kromě podpory pro registry některých periférií obsahuje také linker mapu. Podle té následně linker umísťuje vstupní sekce do výstupních, které korespondují s rozsahy adres pro jednotlivé typy paměti uvnitř mikrokontroléru. Mapa v poskytnuté šabloně však nebrala v úvahu možnost použití USB, pro jehož fungování je vyhrazena vlastní operační paměť statického typu (SRAM). Zprovoznění USB si vynutilo přidání příslušných sekcí po vzoru mapy získané z SDK od výrobce.

Firmware komunikátoru byl vytvořen bez podpory operačního systému (tzv. *"bare metal"*) a koncipován do dvou základních vrstev. První vrstva leží pomyslně blíže k hardwaru a tvoří ji ovladače jednotlivých periférií, potažmo ovladače pro použité integrované obvody a moduly komunikátoru. Druhou vrstvu reprezentuje již samotný aplikační software. Ten využívá rozhraní vrstvy první, která umožňuje abstraktnější přístup k jednotlivým komponentám bez důkladné znalosti jejich struktury. Tyto dvě vrstvy budou ve zbytku kapitoly popsány.

### 6.1 Ovladače

Při realizaci první vrstvy firmwaru byla snaha, alespoň částečně unifikovat vnější rozhraní ovladačů. Jelikož je kladen důraz na energetickou hospodárnost komunikátoru, obsahuje každý ovladač kromě inicializace, také funkce pro povolení a zakázání své činnosti. Nastane-li uvnitř ovladače událost, související s činností spravovaného hardwaru, je volána callback funkce, jejíž

ukazatel je předán při inicializaci. Zvýšení přehlednosti kódu taktéž pomáhá pojmenovávání veřejných prvků tak, že začíná názvem ovladače/knihovny.

### 6.1.1 Ovladač RTC

Periferie RTC uvnitř mikrokontroléru je realizována ve formě 32bitového čítače, který je v případě použití krystalu s frekvencí 32,768 kHz a správném nastavení děličky, navyšován 1x za sekundu. Odlišností od ostatních časovačů je plná funkcionalita v rámci všech režimů, ve kterých může MCU běžet. To umožní nepřerušovaný běh i během restartu MCU. Pro řízení funkcionality slouží osm 32bitových registrů.

První veřejnou funkcí, zajišťující inicializaci, je *RTC\_Init*. Ta nemá návratovou hodnotu a jediným vstupním parametrem je ukazatel na funkci tzv. callback, jež se volá při asynchronní události. Příčinou, v případě ovladače k RTC, může být pouze splnění podmínky alarmu. Předaný ukazatel uvnitř inicializační funkce je uložen do vnitřní statické proměnné a následně v modulu ovládající rozvod hodinového signálu pro periferie (PCC) je povolen přívod. Dalším krokem je povolení přerušování. RTC je k NVIC řadiči připojen skrze multiplexor INTMUX. Ten je nutné povolit a nastavit přerušování na 1. kanál multiplexoru.

Další dvojici inline funkcí tvoří *RTC\_Start* a *RTC\_Stop* ty zajišťují pouze zapsání příslušné hodnoty příznaku povolující funkci RTC uvnitř status registru SR.

Pro interpretaci data a času z číselné hodnoty registru byl využit unixový formát pro datové značky. Jeho hodnota vyjadřuje počet uplynulých sekund od půlnoci 1. ledna 1970. Snadná implementace je dána shodnou šířkou čítacího registru a velikosti datového typu *time\_t*. Funkce pro nastavení aktuálního času *RTC\_SetTime* tedy pouze zapíše hodnotu předanou v parametru do registru s počtem sekund TSR. Funkce pro vyčtení *RTC\_GetTime* naopak formou návratové hodnoty předá aktuální hodnotu TSR.

Poslední vytvořenou funkcí v rámci ovladače RTC je *RTC\_SetAlarm*. Jejím jediným vstupním parametrem je časová značka, při které dojde k vyvolání přerušování, potažmo události.

---

```
void RTC_Init(void (*RTC_event)(void));
void RTC_Start(void);
void RTC_Stop(void);
void RTC_SetTime(time_t time);
time_t RTC_GetTime(void);
void RTC_SetAlarm(time_t time);
```

---

Výpis 1: Veřejné prototypy ovladače RTC

### 6.1.2 LPUART

Sériová asynchronní sběrnice UART, respektive její nízkoodběrová adaptace v podobě LPUART periferie, je komunikačním rozhraním pro rovnocenná koncová zařízení. Jelikož komunikace nemá účastníka, jež by komunikaci řídil, dochází k časově nedeterministickému příjmu. Z těchto důvodů musí být ovladač vybaven dočasným bufferem pro příjem dat. K těmto účelům byla vytvořena knihovna spravující frontu ve formě kruhového bufferu. Tato forma je z hlediska asymptotické složitosti algoritmu velice efektivní (čtení prvního prvku i zápis na konec fronty  $\mathcal{O}(1)$ ).

Rozhraní této knihovny tvoří pětice veřejných funkcí. Protože firmware a všechny jeho komponenty jsou vytvářeny pro použití bez podpory operačního systému, nelze využít žádnou dynamickou alokaci paměti prostřednictvím správce. Je tedy nutné mít již při kompilaci pro frontu vyhrazené dostatečný paměťový rozsah. Aby knihovna mohla být používána univerzálně a nezabírala příliš mnoho paměti zbytečně, je koncipována tak, že při volání inicializační funkce je knihovně předán ukazatel na vyhrazený paměťový prostor z místa použití. To má za následek, že knihovna jen obhospodaruje paměť, která mu byla předána. Dalšími vstupními parametry předávanými při inicializaci je datový typ, maximální počet prvků a také ukazatel na proměnnou, do které bude uložena reference na frontu. Tyto předané údaje jsou uloženy do vnitřní statické struktury, která tyto hodnoty drží pro další použití. Knihovnu je možné používat pro více instancí fronty a pro identifikaci slouží právě předaná reference. Tu je nutné předat pokaždé, když se s frontou pracuje. Funkce pro přidání *Queue\_Enqueue* i vyčtení *Queue\_Dequeue* mají jako druhý parametr obecný ukazatel (typ *void*), přes který se předává zapsaný/vyčtený prvek. Návratovou hodnotu tvoří proměnná výčtového datového, která udává zda funkce skončila úspěšně či nikoliv. Zbytek vnějšího rozhraní tvoří dvě funkce pro zjištění počtu prvku uvnitř fronty (*Queue\_getUsedCount*) a počet volných pozic (*Queue\_getFreeCount*).

---

```
void Queue_Init(QueueHandle_t* handle, void* buffer, uint16_t size,
               Queue_DataType_t type);
Queue_Res_t Queue_Enqueue(QueueHandle_t handle, void* element);
Queue_Res_t Queue_Dequeue(QueueHandle_t handle, void* element);
uint16_t Queue_getFreeCount(QueueHandle_t handle);
uint16_t Queue_getUsedCount(QueueHandle_t handle);
```

---

Výpis 2: Veřejné prototypy knihovny Queue

Periferie LPUART nabízí možnost použití s DMA i konfigurace hardwarové kontroly toku dat. Pro účely této práce však zmíněné funkcionality nebyly potřeba, a proto popisovaný ovladač jejich použití nepřipouští. Obdobná situace nastává v případě nastavení délky datového slova, počtu stop bitu a parity. Zde je na pevně uvnitř ovladače nastavená výchozí konfigurace - počet bitů: 8, stop bity: 1, parita: žádná.



Ovladač nabízí možnost použití 3 kanálů, které korespondují s počtem LPUART periférií uvnitř MCU. Funkce *LPUARTHandler\_Init* provede inicializaci vstupní a výstupní fronty, povolení přerušování, spočtení konstant zajišťující požadovanou symbolickou rychlost a uložení do vnitřní statické struktury. Posledním předávaným parametrem je ukazatel na funkci, která je volána v případě vzniku události. Tu u LPUART vyvolá přijatý bajt, dále přijatý znak pro zalomení řádku, nebo chyba rámce. Návrátová hodnota typu bool značí zda předaná symbolická rychlost je vzhledem k frekvenci hodinového signálu nastavitelná (v rámci 3% tolerance).

Další 2 funkce *LPUARTHandler\_Enable* a *LPUARTHandler\_Disable* pro zapínání a vypínání periférie v rámci různých režimů spotřeby. V případě povolení funkčnosti dojde k přivedení vybraného zdroje hodinového signálu do periférie a následnému nastavení časovacího a kontrolního registru. Zakázání funkčnosti provede softwarový restart periférie a vypnutí taktování.

Funkce pro odeslání dat a odeslání řádku jsou si velmi podobné. Liší se pouze ve svých vstupních parametrech, kde první zmíněné předá ukazatel na data, které mají být poslána spolu s počtem bajtů. V případě poslání řádků se předpokládá, že posílanými daty je textový řetězec. Poté se odešle vše, než ovladač narazí na terminační znak nebo znak značící zalomení řádku. Tyto data jsou vloženy do odesílací fronty a následně je povoleno přerušování ve stavu, kdy je periférie připravena odeslat bajt. V obslužné rutině pak dochází k postupnému vyčítání jednotlivých prvků a zapsání do datového registru. Jakmile dojde k vyprázdnění fronty, je přerušování vypnuto.

Zbývá dvojice funkcí vnějšího rozhraní má podobnou analogii jako ta předchozí.

*LPUARTHandler\_ReadData* slouží k vyčtení zadaného počtu bajtů z vnitřní přijímací fronty. Tyto data jsou uložena do paměti na předané adrese. Poslední funkce *LPUARTHandler\_ReadLine* opět předpokládá, že přijatá data tvoří textový řetězec, a provede vyčtení z fronty takového množství dat, než narazí na výše zmíněné ukončovací znaky.

---

```
bool LPUARTHandler_Init(LPUARTHandler_Chann_t channel, uint32_t
    LPUARTHandler_CLKCFG_PCS, uint32_t srcClkFrq, LPUARTHandler_Baud_t baud,
    void (*LPUARTHandler_Event)(LPUARTHandler_Event_t));
void LPUARTHandler_Enable(LPUARTHandler_Chann_t channel);
void LPUARTHandler_Disable(LPUARTHandler_Chann_t channel);
LPUARTHandler_Res_t LPUARTHandler_SendLine(LPUARTHandler_Chann_t channel, char
    data[]);
LPUARTHandler_Res_t LPUARTHandler_SendData(LPUARTHandler_Chann_t channel, void*
    data, uint16_t len);
void LPUARTHandler_ReadData(LPUARTHandler_Chann_t channel, void* data, uint16_t
    len, uint16_t* recLen);
```

```
void LPUARTHandler_ReadLine(LPUARTHandler_Chann_t channel, void* data, uint16_t  
    maxLen, uint16_t* recLen);
```

---

Výpis 3: Veřejné prototypy ovladače LPUART

### 6.1.3 Wi-Fi RN-131

Vybraný Wi-Fi modul RN-131 je k mikrokotroléru připojen pomocí sběrnice UART. Dříve popsaný ovladač tedy s výhodou využijeme. Jelikož celý modul je velmi komplexní se spoustou funkcionalit a nastavení, je ovladač vytvořen pouze v rozsahu použití potřebné pro tuto práci.

Ovladač Wi-Fi modulu předpokládá použití výhradně jako klienta v infrastrukturním režim, kdy připojením k AP získá přístup na internet. Dalším předpokladem před vytvořením ovladače bylo použití GET metody pro předávání dat v rámci HTTP požadavku pro komunikaci se vzdáleným serverem. Rovněž předávání dat je pouze jednostranné, server při zdařilém zpracování přijatých dat posílá zpět pouze potvrzovací řetězec. Modul RN-131 má 2 základní režimy. Tím první je komunikační, kdy přichází data po UARTu jsou ukládána do odesílacího bufferu. Druhý režim je příkazový. V něm se využívá textových příkazů pro nastavování parametrů, konfiguraci a ovládání chování Wi-Fi modulu. Výhodou je možnost uložení veškerého nastavení.

Tak jako u všech dosud popsaných ovladačů, tak i u tohoto je první veřejnou funkcí inicializace. Při volání se formou vstupního argumentu předá ukazatel na callback funkci. V případě Wi-Fi modulu ovšem její vyvolání může způsobit více událostí. Rozlišit důvod, proč byl callback vyvolán lze na základě vstupního parametru funkce ve formě výčtového datového typu. Dalším krokem, který se v rámci inicializace provádí je nastavení kanálu 0 INTMUXu pro LPUART2, povolení přerušení, inicializace fronty pro příkazy a ovladače pro LPUART2. Druhou standardní funkci v rámci vytvořených ovladačů je povolení činnosti Wi-Fi modulu *rn131\_Enable*. Zde dojde k aktivaci GPIO výstupu pro otevření tranzistoru, jež spíná napájení a povolení UARTu. Vstupním argumentem se udává, zda má ovladač pracovat v běžném režimu, kdy plně obsluhuje komunikaci s modulem, nebo zda má pouze sloužit jako prostředník při předávání dat na/z UART sběrnice. Funkce pro zakázání činnosti deaktivuje GPIO výstup zajišťující napájení a rovněž vypne LPUART periferii.

Obecně platí, že zpracovávání textové komunikace, je výpočetně náročnější než je tomu v případě binárního protokolu a jelikož komunikace s Wi-Fi modulem je výhradně textová, byla vytvořena funkce *rn131\_Service*. Ta slouží jako služba, která musí být cyklicky volána z hlavní programové smyčky. Jelikož obslužné rutiny přerušení musí být co nejefektivnější, aby nedocházelo k nadměrnému zatěžování asynchronním vykonáváním programu, jsou přichází data pouze ukládána do přijímacího bufferu a časově náročné zpracování zajišťuje zmíněná služba. Uvnitř služby je implementován stavový automat, který řídí přechody mezi komunikačním a příkazovým

režimem, vykonávání příkazů a odesílání dat. Aby se rozhraní ovladače dalo udělat co nejabstraktnější a oddělené od skutečného fungování modulu, byla vytvořena fronta pro posloupnosti příkazů. Například skenování dostupných Wi-Fi sítí vyžaduje přepnutí do příkazového režimu, následně odeslání skenovacího příkazu *scan*, po dokončení zpracovat vrácený seznam sítí do připravené struktury a přejít zpět do komunikačního režimu příkazem *exit*. Pro toto vše stačí z nadřazené vrstvy zavolat funkci *rn131\_Scan*. Ta zajistí naplnění fronty příslušnými příkazy a všechny ostatní úkony jsou vykonány již automaticky v rámci popisované služby. Jakmile dojde k přepnutí zpět do komunikačního režimu, je nadřazená vrstva informována prostřednictvím vyvolané události.

Dvojice funkcí *rn131\_SendRAW* a *rn131\_ReadRAW* je určená pro odesílání, respektive vyčítání surových dat z UART sběrnice v RAW módu. Jak bude později popsáno, toto je využito v rámci Wi-Fi tunelového režimu komunikátoru.

*rn131\_IsJoined* je jednoduchá funkce, která svou návratovou hodnotou říká, zda je modul úspěšně připojen k nějaké AP, či nikoliv. Obdobně *rn131\_IsIdle* vrací hodnotu typu bool, jež dává informaci o tom, zdali je stavový automat v režimu připravenosti a nevykonává momentálně žádnou sérii příkazů.

Další důležitou funkcí z vnějšího rozhraní je *rn131\_SendData*, ta v případě dostupného internetového připojení provede odeslání dat pomocí HTTP požadavku. Wi-Fi modul nabízí 3 možnosti jak aktivovat odeslání dat, jež má v bufferu. První z nich je nastavení velikostí bufferu, jakmile je naplněn, dojde k odeslání dat. Druhou cestou je využití časovače, do něj se nastaví maximální doba čekání na další bajt a po jejím uplynutí jsou odeslána nashromážděná data. Třetí a také použitou možností je vyhradit jednu bajtovou hodnotu jako zakončovací znak pro odeslání. V případě této diplomové práce byl vyhrazen znak 'k' (0x6B). Problém ovšem nastává v případě, že se taková bajtová hodnota nachází v užitných datech. Modul RN-131 zároveň nedovoluje odesílat GET metodou některé bílé znaky. To neodvratně vede k vytvoření komunikačního protokolu, který tyto náležitosti zohlední. Pro jednoduchost byla zvolena možnost textového odesílání hexadecimální hodnoty jednotlivých bajtů. Odeslaný rámec má poté dvojnásobnou velikost oproti užitným datům. V případě Wi-Fi připojení komunikátoru to však nečiní problémy, jelikož tyto sítě zpravidla nebývají omezovány FUP limity jako je tomu v případě GSM. Odesílací funkce tedy naformátuje předaná data výše popsáním způsobem a pošle skrze UART. Jakmile server data zpracuje, odpoví zda všechno proběhlo v pořádku nebo ne. Tato informace je nadřazeném systému předána formou události. Služba rovněž měřením časové prodlevy odpovědi hlídá, zda nedošlo k chybě, což by způsobilo případné zamrznutí.

Dojde-li k vyčtení nastavení modulu nebo skenování Wi-Fi sítí jsou výsledky uloženy do vnitřních statických proměnných. Pro přístup k nim slouží funkce *rn131\_GetStatusVars*, respektive

*rn131\_GetScanRes*. Ty předají ukazatel na strukturu, pomocí níž lze přistupovat k jednotlivým údajům.[18]

---

```
void rn131_Init(void (*RN131_Event)(RN131_Event_t));
bool rn131_Enable(bool raw_mode);
void rn131_Disable(void);
void rn131_Service(void);
bool rn131_SendRAW(void *data, uint16_t len);
bool rn131_ReadRAW(void **data, uint16_t *recLen);
bool rn131_IsJoined(void);
bool rn131_IsIdle(void);
bool rn131_Join(void);
bool rn131_Scan(void);
bool rn131_ReadStat(void);
bool rn131_SendData(void *data, uint8_t len);
RN131_Wifi_t* rn131_GetStatusVars(void);
RN131_Wifi_ScanRes_t* rn131_GetScanRes(void);
```

---

Výpis 4: Veřejné prototypy ovladače Wi-Fi modulu

#### 6.1.4 IQRF

IQRF moduly nabízejí na základě nahraného hardwarové profilu HWP možnost komunikace buď pomocí sběrnice UART nebo SPI. Z důvodu snadnějšího vytvoření tunelového IQRF režimu komunikátoru byl zvolen HWP koordinátora s podporou UARTu. Při použití tohoto rozhraní je však nutné navíc implementovat na linkové vrstvě v rámci ISO/OSI modelu HDLC protokol. Jelikož při asynchronní komunikaci může mezi jednotlivými bajty být nedefinovaný časový rozestup, je výhodné používat protokol, který zajistí rozpoznání začátku a konce rámce. IQRF tedy vyžaduje, aby každý DPA paket byl zabalen do rámce prostřednictvím zmíněného HDLC protokolu. Softwarová implementace byla převzata z výzkumného projektu [11].

Vnější rozhraní ovladače IQRF modulu má 7 funkcí. První 3 tvoří standardně inicializaci, povolení a zakázání činnosti. V rámci *IQRF\_Init* dojde k inicializaci ovladače UARTu a knihovny HDLC. Funkce pro povolení činnosti zajistí přivedení napájení do modulu a spustí skrze ovladač periférii LPUART0. Funkce *IQRF\_Disable* naopak deaktivuje funkčnost komunikačního rozhraní a napájení vypne.

Zbylé 4 funkce, které ovladač nabízí slouží k odesílání a příjmu dat. Alokace paměti pro vnitřní buffer ovladače musela zohlednit, že jeden DPA paket ze své definice může mít maximálně 64 bajtů. Rámec vytvořený HDLC protokolem zase 130 bajtů. Pro tunelový Wi-Fi režim jsou určeny

vytvořené funkce *IQRF\_SendRAW* a *IQRF\_ReadRAW*, ty zajistí odeslání/příjem předaného rámce, bez zásahu HDLC algoritmu. Naproti tomu funkce *IQRF\_SendDPA* předpokládá předání DPA paketu, z něhož je vytvořen rámec, jež je následně odeslán. Funkce *IQRF\_ReadDPA* slouží pro vyčtení příchozího rozbaleného paketu.

Součástí ovladače je také časovač, který hlídá dobu odezvy modulu. Je-li poslán dotaz na měřicí uzel, který je však vypnut, přichází pouze od koordinátora potvrzení, avšak odpověď již nepřijde. V tomto případě vyprší hodnota časovače a ten nadřazenou vrstvu informuje příslušnou událostí.[15]

---

```
void IQRF_Init(void (*IQRF_Event)(IQRF_Event_t));
bool IQRF_Enable(void);
bool IQRF_Disable(void);
bool IQRF_SendDPA(uint16_t naddr, uint8_t pnum, uint8_t pcmd, uint16_t hwpid,
                  uint8_t *pdata, uint16_t pdata_len);
bool IQRF_SendRAW(void *data, uint16_t len);
bool IQRF_ReadDPA(IQRF_DPA_Res_t **reponse, uint8_t *pdata_len);
bool IQRF_ReadRAW(void **data, uint16_t *recLen);
```

---

Výpis 5: Veřejné prototypy ovladače IQRF modulu

### 6.1.5 FRAM

Vybraná feroelektrická paměť je připojena pomocí komunikační sběrnice SPI. Jelikož pro žádnou další komponentu nebylo toto rozhraní využito, je ovladač paměti vytvořen konkrétně pro daný MCU. Periferie LPSPI nabízí velké množství nastavení pro fungování jak v roli master, tak slave. Pro odesílání i příjem je periferie vybavena hardwarovou frontou, která umožňuje dávkové plnění/odebírání přenesených dat, přičemž maximální velikost rámce je 256 bajtů. Komunikace s pamětí je realizována na frekvenci 1 MHz. Vnější rozhraní ovladače je navrženo pro co nejjednodušší použití. Standardní funkce pro inicializaci, povolení a zakázání doplňuje *FRAM\_Write*, *FRAM\_Read*. Ty slouží pro zápis a vyčítání dat z paměti. Mají rovněž stejné vstupní parametry, kterými jsou adresa paměti, délka dat a ukazatel na místo odkud/kam mají být data zapsána/přečtena z FRAM. Pro případ, že nadřazená vrstva chce pracovat s blokem dat větším než 256 bajtů, by musel být implementován mechanismus, který provádí čtení i zápis dat automaticky po částech.

Pro ovládání je vytvořen systém příkazů, tzv. opcode. Tyto příkazy jsou zasílány vždy na začátku transakce a stanovují co se po paměti aktuálně požaduje. Pro zápis je nutné nejprve odeslat příkaz povolující zápis a následně spolu s adresou WRITE opcode. Po této režii již následují data, jejichž ukládání započne na odeslané adrese. Čtení má obdobný scénář s výjimkou, že není

nutné odesílat žádný povolovací příkaz, ale stačí pouze opcode pro čtení za nimž již FRAM odesílá data, která má uložena.[17]

---

```
void FRAM_Init(void (*FRAM_Event)(FRAM_Event_t));
bool FRAM_Enable(void);
bool FRAM_Disable(void);
bool FRAM_Write(uint32_t addr, void* data, uint16_t len);
bool FRAM_Read(uint32_t addr, void* data, uint16_t len);
bool FRAM_ReadStat(uint8_t* data);
```

---

Výpis 6: Veřejné prototypy ovladače FRAM paměti

### 6.1.6 Bateriový monitor

V kapitole věnující se návrhu hardwaru již bylo zmíněno, že vybraný integrovaný obvod BQ28Z610 umožňuje prostřednictvím sběrnice I<sup>2</sup>C vyčítat sadu registrů, jež obsahují jak měřené, tak spočtené hodnoty týkající se stavu akumulátoru. Společným rysem je vybavenost periferií LPI2C a LPSPi 4prvkovou frontou pro transakce. Rozhraní ovladače pro bateriový monitor obsahuje 6 funkcí. Tři z nich tvoří standardní inicializace, povolení a zakázání činnosti. Funkce *BATMON\_ReadState* zajišťuje jednorázové vyčtení sady registrů a uložení hodnot do vnitřní struktury stavu. Obdobnou funkcionalitu plní *BATMON\_StartReading*, avšak ta využívá pro cyklické vyčítání hodnot hardwarový časovač LPIT0. Vstupním argumentem je perioda obnovy v sekundách. Poslední funkci je jednoduché předání ukazatele na stavová data (SOH, SOC, napětí a proud akumulátoru) reprezentované strukturou.[16]

---

```
void BATMON_Init(void (*BATMON_event)(BATMON_Event_t event));
bool BATMON_Enable(void);
bool BATMON_Disable(void);
bool BATMON_ReadState(void);
bool BATMON_StartReading(uint8_t period_s);
BATMON_Status_t *BATMON_GetStatus(void);
```

---

Výpis 7: Veřejné prototypy ovladače bateriového monitoru

### 6.1.7 USB

Použití USB rozhraní je díky své univerzálnosti poměrně složité. Každé koncové zařízení se svému hostiteli po připojení identifikuje (třídou, VID, PID), načež hostitel provede inventarizaci a rozdělení adres. Na základě přidělených adres jsou vytvořeny virtuální roury (pipes), na jejichž koncích se nachází tzv. endopinty. Výhodou je, že o sdílení sběrnice mezi rourami se stará samotné USB a ovladače na jednotlivých stranách již pak mohou ke komunikaci přistupovat jako by se jednalo o exkluzivní P2P spojení.

Pro účely komunikátoru bylo zvoleno použití USB jako virtuální sériové linky. Výhodou tohoto pojetí je vybavenost běžně používaných operačních systémů počítačů generickými ovladači, které se k připojení zařízení automaticky asociují. Aby byla zachována tato kompatibilita, byl ovladač převzat z SDK od výrobce KL28Z512ZVLL7. Jelikož se však předpokládá použití celého SDK, musely být z kódu odstraněny ostatní knihovny, které s USB nesouvisí, např. pro nastavení rozvodu hodinového signálu. Další problém způsoboval generický ovladač na OS Windows, ten v rámci režijní komunikace nenastavuje příznak značící začátek transakce. Pro následné použití bylo vynucování tohoto příznaku potlačeno.

Rozhraní ovladače obsahuje dvojici funkcí pro inicializaci a deinicializaci. V rámci první zmíněné se jako vstupní argument předává ukazatel na callback funkci. Stejně jako v případě Wi-Fi modulu, tak i zde je nutné cyklicky volat funkci *USB\_VCOM\_Service*. V tomto případě však uvnitř nedochází k žádnému zpracovávání textové komunikace, ale slouží jen pro odesílání dat z bufferu a odpovídání na servisní zprávy. Následující funkce *USB\_VCOM\_isReady* formou své návratové hodnoty vrací informaci, zdali je USB řádně připojeno k hostiteli. Další funkce již slouží pro samotné zapsání/čtení z/do odesílacího/přijímacího bufferu. Poslední funkce *USB\_VCOM\_getRecSize* pak vrací aktuální počet bajtů připravených pro přečtení.[4]

---

```
bool USB_VCOM_Init(void (*USB_VCOM_Event)(void*, uint32_t, USB_VCOM_Event_t));
void USB_VCOM_Deinit(void);
bool USB_VCOM_Service(void);

bool USB_VCOM_isReady(void);

bool USB_VCOM_ReadAll(void* data, uint32_t* len);
bool USB_VCOM_ReadData(void* data, uint32_t len);
bool USB_VCOM_SendData(void* data, uint32_t len);

uint32_t USB_VCOM_getRecSize(void);
```

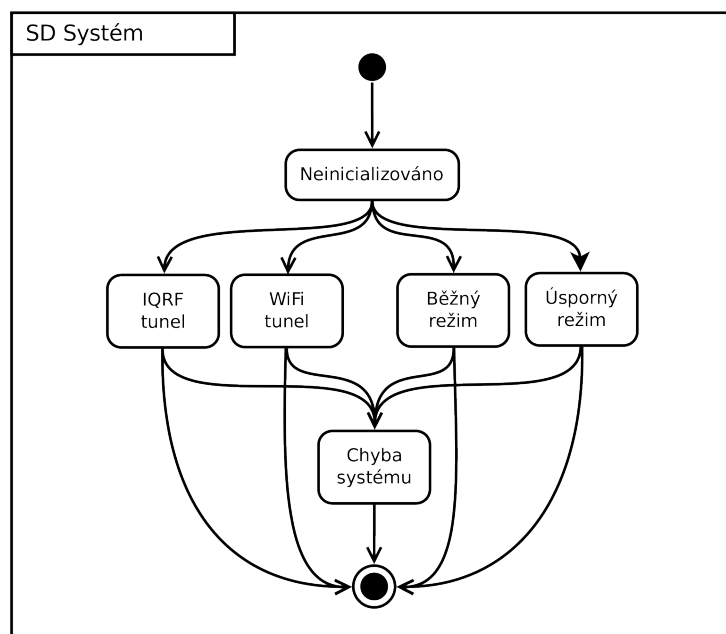
---

Výpis 8: Veřejné prototypy ovladače USB

## 6.2 Systém

Aplikační software, jež odpovídá druhé vrstvě zmíněné v úvodu kapitoly, se fakticky dělí na 4 částí, které korespondují s funkčními režimy. Tunelové módy slouží pouze pro servisní připojení k jednotlivým komunikačním modulům. Běžný režim již slouží pro sběr a odesílání dat, navíc je po připojení k počítači možné provádět pomocí servisního protokolu úpravy nastavení. Energeticky úsporný režim nabízí pouze možnost sběru dat.

Pro každý z režimů byl vytvořen separátní program, který má svou inicializační a cyklickou funkci. Po spuštění mikrokontroléru dojde k aktivaci RTC a GPIO vstupně výstupních portů, ke kterým jsou připojeny indikační LED diody, tlačítko a DIP přepínač. Následně je podle kombinace na DIP přepínači spuštěn příslušný režim.



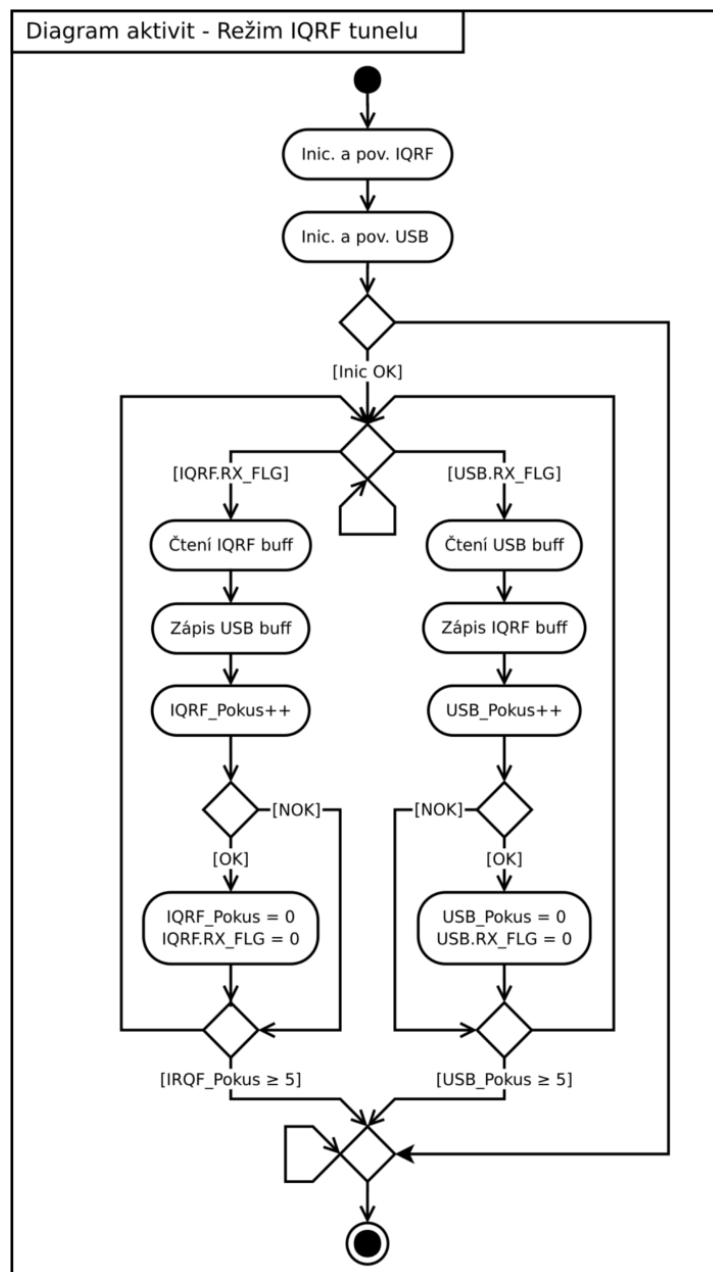
Obrázek 33: Stavový diagram systému komunikátoru

### 6.2.1 Režim IQRF tunelu

Vývojové prostředí IQRF IDE nabízí možnost komunikačního spojení s modulem buď prostřednictvím programátoru (SPI sběrnice) nebo pomocí sériové linky doplněné o HDLC protokol. Zatímco v případě spojení přes SPI je vyžadován HW od firmy Microrisc, který se v operačním systému počítače registruje jako kompatibilní zařízení, v případě komunikace přes sériovou linku lze využít kterýkoliv COM port. Jako přirozené se tedy jevílo použít v tomto režimu MCU jako prostředníka, který bude data pouze předávat z USB na UART a naopak.

Po úspěšném spuštění tohoto režimu dojde k inicializaci a povolení ovladače pro IQRF a USB. Jestliže proběhly tyto funkce bez chyb, přejde program do stavu čekání na příchozí data. Jakmile ovladač jedné nebo druhé periferie přijme zprávu, dojde k vyzvednutí a následnému předání dat druhému ovladači. V případě že by na sběrnici nastala chyba a data se nepodařilo odeslat, implementovaný firmware zajistí opětovné odeslání. Pakliže by nastala trvalá chyba, a ani 5. pokus by nebyl úspěšný, přejde komunikátor do chybového stavu.



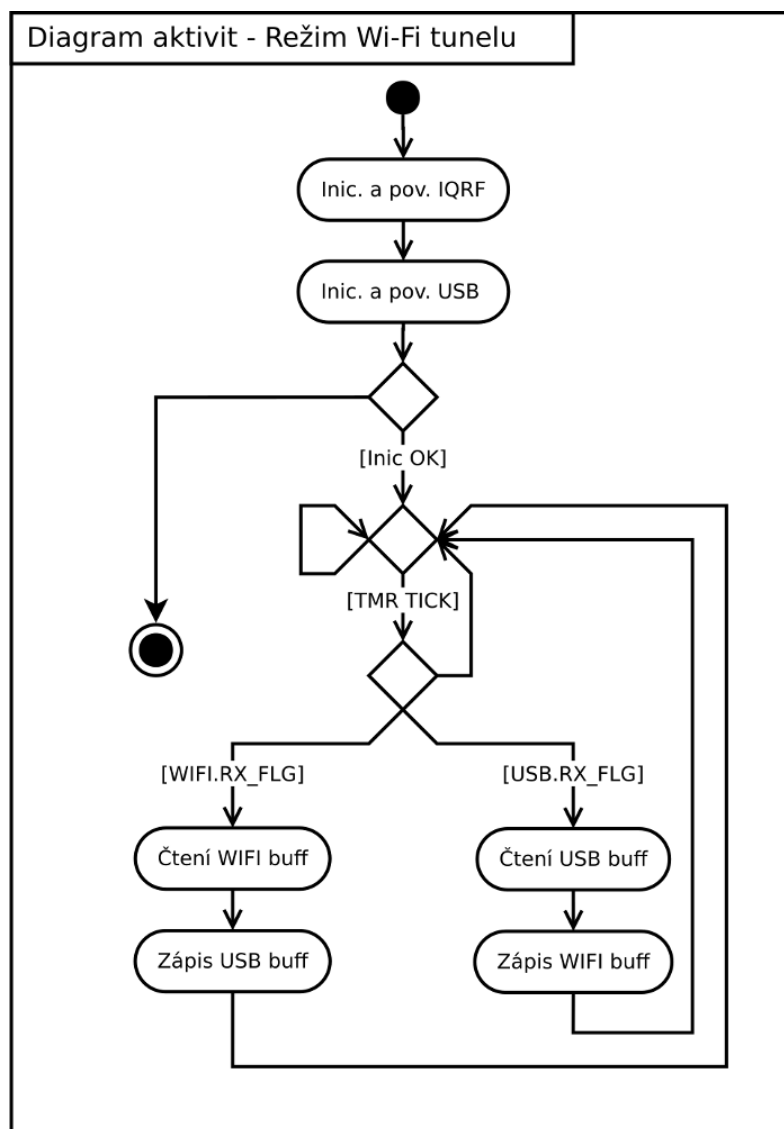


Obrázek 34: UML diagram aktivit pro režim IQRF tunelu

### 6.2.2 Režim Wi-Fi tunelu

Režim Wi-Fi tunelu je analogický s výše zmíněným. Wi-Fi modul používá pro nastavení a konfiguraci textových příkazů, které jsou uzpůsobeny pro použití společně se sériovým terminálem. V rámci tohoto režimu tedy mikrokontrolér vystupuje v roli převodníku z USB na UART. Po připojení komunikátoru k počítači se mikrokontrolér registruje v operačním systému jako sériová linka. Tento port je poté možný vybrat v sériovém terminálu a komunikovat tak s modulem, jako by byl připojen napřímo skrz převodník.

Oproti předchozímu režimu byl zde implementován softwarový časovač. Ten využívá čítací dělicí registr od RTC modulu a zajišťuje periodu přeposílání dat. Tato funkcionality byla implementována z toho důvodu, že sériový terminál posílá každý znak v samostatné zprávě přes USB. Odesílací perioda byla zvolena  $\frac{1}{16}$  sekundy. Po jejím uplynutí dojde ke kontrole naplněnosti bufferu, jsou-li přijata nějaká data, jsou předána na druhé rozhraní.



Obrázek 35: UML diagram aktivit pro režim Wi-Fi tunelu

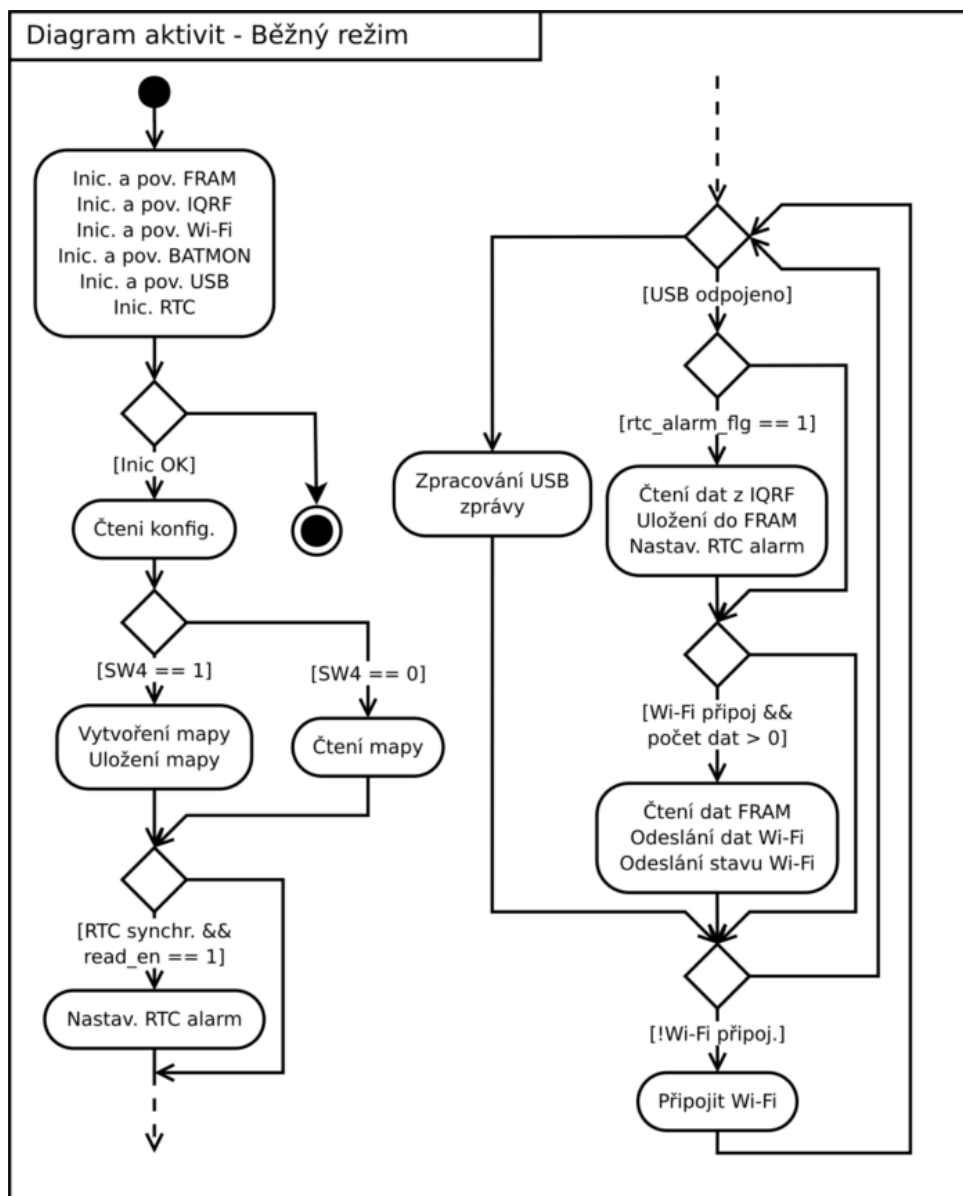
### 6.2.3 Bežný režim

Tento režim má všestranné použití. Umožňuje sběr a odesílání dat rovněž s možností připojení k počítači skrz USB pro účely konfigurace. V rámci nastavení je v energeticky nezávislé paměti uložena časová značka začátku měření a také měřicí interval. Z těchto údajů je vždy s použitím aktuální hodnoty RTC spočten čas následujícího měření.

Aby bylo možné používat automatické vytváření mapy IQRF sítě pro účely měření, je nutné zachovat domluvené parametry DPA dotazů. IQRF nabízí v rámci tzv. *User DPA handleru* možnost vytvoření virtuálních periférií. Při odeslání požadavku na tuto periférii je vykonán uživatelský program IQRF modulu a zasláná odpověď. To vedlo k nutnosti stanovit pro funkčnost měřicího zařízení s navrženým komunikátorem pevně PNUM na hodnotu 0x20 a PCMD na 0x31.

Po spuštění běžného režimu dojde k inicializaci a povolení ovladačů všech komponent integrovaných na DPS komunikátoru. Dalším krokem je přečtení konfigurace z paměti FRAM a spuštění cyklického vyčítání stavu akumulátoru. Poté dojde k vyčtení hodnoty čtvrtého konfiguračního DIP přepínače. Ten určuje, zda má být použita mapa IQRF sítě, jež je uložena v energeticky nezávislé paměti nebo zda se má vytvořit nová. Vytvoření nové mapy začíná zasláním dotazu koordinátorovi. Ten mikrokontroléru zašle adresy zaregistrovaných ("*přibondovaných*") měřicích uzlů. Podle této získané tabulky se komunikátor postupně dotazuje všech uzlů na výrobní číslo MID a data, která měří. Tyto údaje jsou následně využity k sestavení mapy, která pro každý uzel obsahuje adresu NADR, výrobní číslo MID, hardwarový identifikátor HWPID a počet měřených bajtů. Byla-li mapa vytvořena korektně, je pro další použití zapsána do paměti FRAM. Dalším krokem při náběhu běžného režimu komunikátoru je zjištění, zdali byly hodiny reálného času synchronizovány. Tato kontrola spočívá v porovnání aktuální hodnoty čítacího registru s konstantou odpovídající 1.1.2019. Je-li mikrokontrolér opětovně spuštěn po odpojení napájení, nachází se v registru výchozí hodnota odpovídající roku 1970. Při vyhodnocení neprovedené synchronizace je rozsvěcena červená LED dioda a zamezeno sběru dat z IQRF sítě, jelikož by nebylo možné vytvořit naměřeným hodnotám časovou značku. Po skončení kontroly přejde firmware do stavu připravenosti. V něm probíhá cyklická kontrola příznaku, zdali není komunikátor připojen pomocí USB k počítači. Je-li tomu tak, dojde k dočasné deaktivaci sběru a odesílání dat. Toto opatření je nutné z důvodu možné změny konfigurace a zajištění exkluzivního přístupu k IQRF a Wi-Fi modulům z části programu, jež obhospodaruje přijaté zprávy přes USB. Pokud není komunikátor připojen k PC, přechází k čekání na příznak od alarmu RTC, zároveň se vyhodnocuje, zda jsou uloženy nějaké měření v datovém zásobníku. Při nastalém RTC alarmu přejde program ke sběru dat z IQRF sítě. Podle mapy se komunikátor dotazuje na data jednotlivých uzlů. Došlé data jsou uloženy do zásobníku paměti FRAM. V případě neúspěšné žádosti, je dotaz opakován, maximálně však 5x. Nejsou-li komunikátoru zaslána požadovaná data, je uzel přeskočen. Po dokončení měření se provede kontrola zda je zásobník prázdný. Pokud není a navíc je komuni-

kátor řádně připojen dojde k postupnému odeslání nashromážděných dat. V neposlední řadě se v režimu připravenosti zajišťuje připojení k Wi-Fi síti.

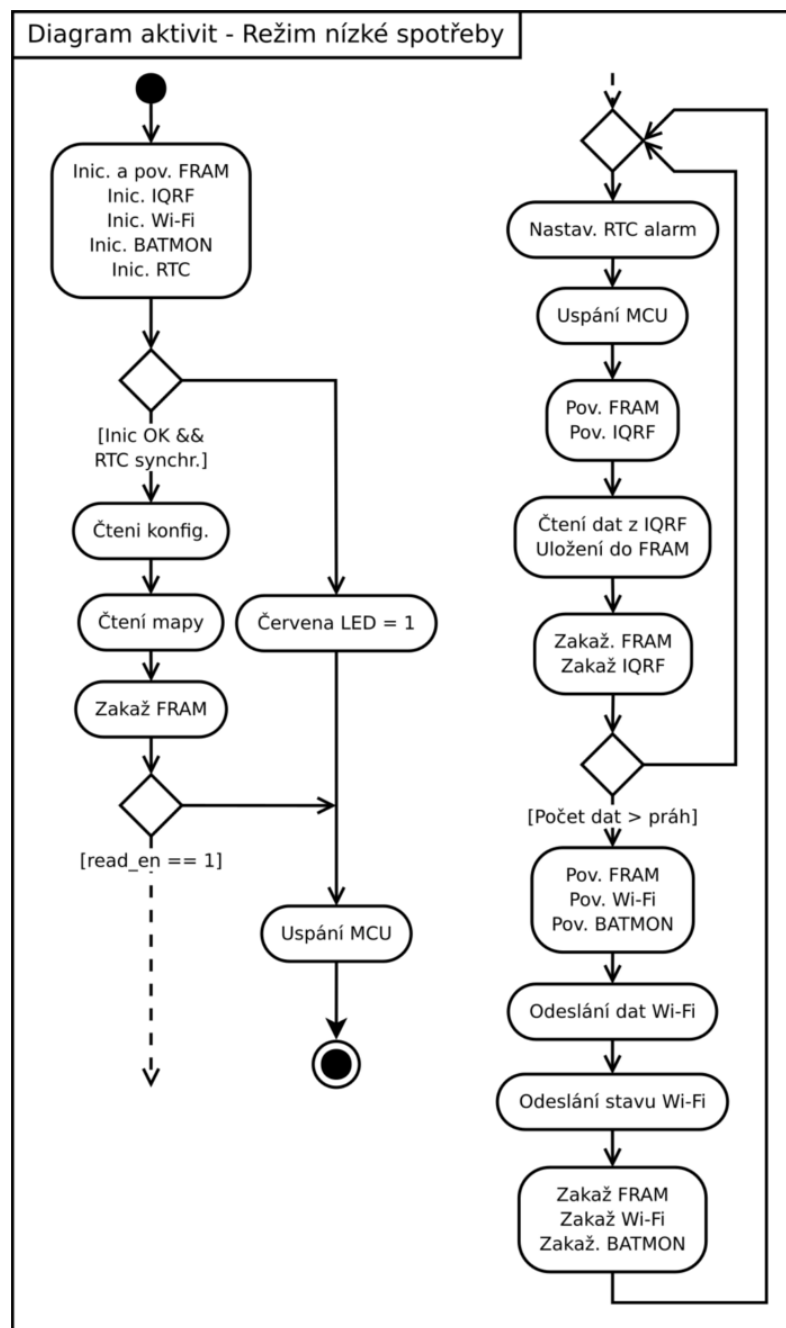


Obrázek 36: Zjednodušený UML diagram aktivit pro běžný režim

#### 6.2.4 Režim nízké spotřeby

Funkčnost v tomto režimu je omezena pouze na sběr a odesílání dat na vzdálené internetové úložiště, přičemž je minimalizovaná spotřeba energie. Toho je docíleno jednak vypnutím všech komponent, které aktuálně nejsou potřeba, dále usmáním procesorového jádra, ale také spouštění Wi-Fi modulu a následné odesílání až při nasbírání většího množství dat.

Po spuštění režimu nízké spotřeby dojde k inicializaci ovladačů všech komponent s výjimkou USB, jehož použití je v tomto módu zapovězeno, a následně je povolena činnosti energeticky nezávislé paměti FRAM. Proběhly-li všechny inicializační funkce korektně a zároveň je vyhodnoceno, že hodiny reálného času jsou synchronizovány, dojde k vyčtení konfigurace a mapy sítě z FRAM paměti. V opačném případě dojde k rozsvícení červené indikační LED diody na znamení chyby a mikrokontrolér je uspán bez možnosti probuzení. Je tomu tak z důvodu, že jedinou funkcionalitou úsporného režimu je sběr a odesílání dat, což není možné, jelikož komunikátor nedokáže datům přidělovat časové značky. Je-li však kontrola i vyčtení konfigurace spolu s mapou úspěšné, dojde k vypnutí FRAM. Následně se z načtené konfigurace vyhodnotí, zda je příznak pro zapnutí sběru dat roven logické 1 nebo ne. Pokud není, dojde opět k trvalému usnutí. Jakmile je ovšem sběr dat povolen, provede se nastavení alarmu RTC a mikrokontrolér přejde do módu STOP. Z množství nabízených režimů spánku je STOP výhodný, jelikož po provedení instrukce WFI, přejde jádro do hlubokého spánku, hodiny pro periferie jsou vypnuty, ale zároveň zůstávají zachovány hodnoty ve všech systémových registrech. Probuzení je zajištěno přerušením přivedeným do AWIC (Asynchronous Wake-up Interrupt Controller). V případě komunikátoru je jako zdroj využit alarm RTC. Díky uchované hodnotě systémových registrů program pokračuje následující instrukcí po WFI, čili od místa, kde byl uspán. Po úspěšném probuzení program pokračuje povolením činnosti FRAM a IQRF. Z IQMESH sítě jsou podle mapy postupně vyčtena data a uložena do zásobníku, následně dojde k vypnutí zmíněných komponent. Pro další energetickou úsporu, byla do firmwaru komunikátoru implementována možnost nastavit prahové množství naměřených dat, kdy až po překročení tohoto prahu dojde k odeslání celého balíku dat najednou. Například měření může probíhat co hodinu, ale odeslání dat ze zásobníku na datové úložiště se provede pouze jednou za den. Tedy při překročení prahového množství měření, dojde k povolení FRAM, Wi-Fi a bateriového monitoru. Ze zásobníku jsou postupně vyčítány hodnoty a odesílány na serverové úložiště. Jakmile je přenos dat hotov, komunikátor se dotáže bateriového monitoru na stav akumulátoru. Stupeň nabití SOC je následně odeslán na server, kde se udržují informace o komunikátoru z doby, kdy byl naposledy připojen. Po všech těchto krocích se provede nastavení alarmu RTC na další měření a mikrokontrolér se znovu usní.

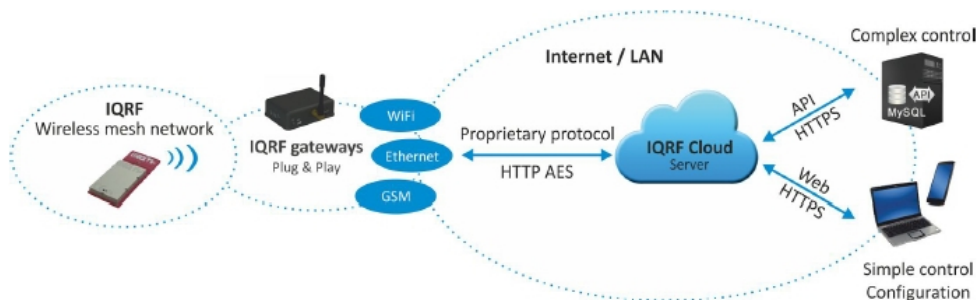


Obrázek 37: Zjednodušený UML diagram aktivit pro režim nízké spotřeby

## 7 Popis možností a realizace přenosu dat z IQRF sítě na datové úložiště

Za účelem snadné dostupnosti nasbíraných dat z IQRF bezdrátové MESH sítě pro uživatelské systémy byl vystavěn IQRF Cloud server. Poskytovatelem této služby je smluvní partner společnosti IQRF Tech s.r.o. firma CIS, s.r.o. Tímto propojením se stává IQRF zařízení součástí koncepce IoT, kde nachází široké uplatnění např. v telemetrii nebo v průmyslovém řízení.

Značnými výhodami IQRF Cloudu je nejenom výše zmíněný snadný přístup k datům, ale i možnost jejich zpracování a vizualizace. Předností této služby je i zabezpečení získaných dat a jejich bezpečný přenos, který mezi IQRF Cloudem a uživatelskou aplikací probíhá prostřednictvím HTTPS protokolu. Mezi podporované uživatelské platformy pro využití dat z IQRF Cloudu patří standardní nástroje jako například PHP, JavaScript nebo webové rozhraní k jehož použití však musí být vytvořen uživatelský účet. IQRF Cloud také poskytuje několik možností, jakými mohou být data z něj vyčítány. První z nich je prvotní čtení dat, další mi možnostmi je čtení dat na základě jejich indexu nebo času a poslední variantou je čtení požadovaného množství nasbíraných dat.



Obrázek 38: Schématické znázornění přenosu dat z IQMESH sítě pomocí IQRF brány

Propojení IQMESH sítě s cloudovým úložištěm je zprostředkováno pomocí IQRF bran, které je možno rozlišit na základě požadovaného spojení – Ethernet, Wi-Fi nebo GSM. Primární směr toku dat je dopředný, tedy předávání dat z brány do IQRF. Pro nastavení parametrů nebo doručení příkazů do koncového zařízení je navíc umožněn zpětný tok dat, tedy opačný pohyb dat z IQRF Cloudu do brány. Tato obousměrná komunikace probíhá dle určeného neveřejného protokolu, což má značný negativní vliv na použití IQRF Cloudu třetím zařízením, které se stávají tímto nekompatibilním. Tyto brány, pak také mohou sloužit pro vzdálený přístup do IQMESH sítě z IQRF IDE. Je-li brána připojena k internetu, musí mít pro možnost vzdálené správy přidělenou veřejnou IP adresu. Za tyto adresy si však obvykle poskytovatelé účtují další poplatky.

Nově vznikající a expandující trh se zařízeními a službami pro IoT se snaží využít i další společnost, které poskytují cloudové služby. Mezi nejrychleji rozvíjející se patří Google Cloud, Microsoft Azure, IBM Watson IoT nebo Amazon AWS IoT Cloud. Všechny tyto služby jsou poskytovány za úplatu, jejíž výše se zpravidla odvíjí od velikosti využití diskové kapacity nebo od množství přenesených zpráv. Spojení těchto cloudových služeb s IQMESH sítí lze buď pomocí představených bran spolu s IQRF Cloudem, přes který budou data přeposílána nebo pomocí přímého spojení skrze zařízení třetí strany. Jedním z možných řešení je využít Raspberry PI s IQRF modulem, na kterém zpravidla běží linuxový operační systém. Pro něj je dostupný IQRF GW daemon, který spolu se softwarem Node-RED je schopen vytvořit výkonnou bránu pro spojení IQMESH sítě s internetovým úložištěm. Komunikace pro potřeby IoT je často realizována prostřednictvím protokolu MQTT. Tato řešení však nejsou příliš vhodná pro nízkoodběrové aplikace.

## 7.1 Realizace datového úložiště

Prvotním cílem při řešení této práce bylo připojení navrženého komunikátoru na IQRF Cloud, avšak firma CIS, s.r.o. tuto eventualitu zamítla s tím, že připojením je možné pouze pro OEM zařízení. Jelikož vybraný Wi-Fi modul nemá vestavěnou podporu pro MQTT protokol a bylo by jej nutné implementovat na úrovni mikrokontroléru s využitím TCP klienta, bylo vytvořeno vlastní datové úložiště, které využívá HTTP protokol a MySQL databázi.

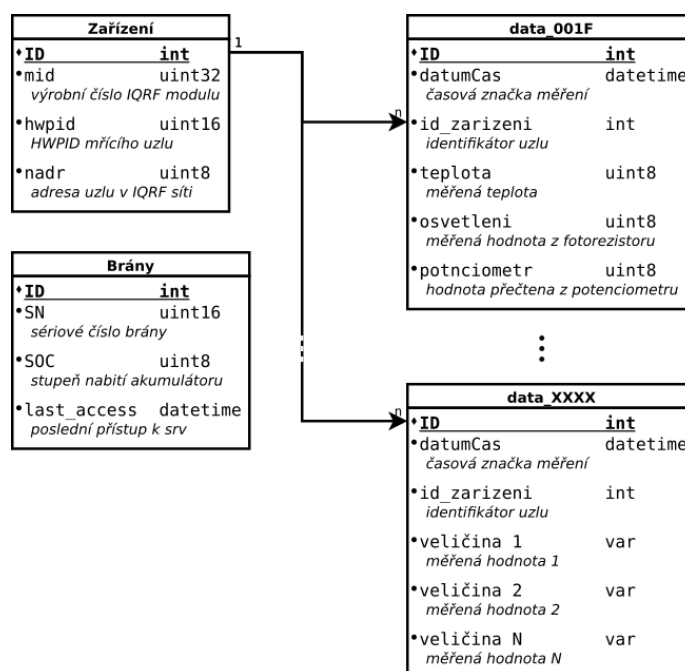
Tuto kombinaci často využívají webové servery a jako přirozená možnost se tedy jevílo využít nabídku některého z poskytovatelů webhostingu. Ti zpravidla nabízejí pronájmy zdarma s omezením v podobě nízké velikosti databáze a limitem pro přenesená data. Využita byla nabídka firmy Stable.cz s.r.o., jež provazuje servery Endora.cz. Ta nabízí webhosting zdarma s limitem velikosti databáze 30 MB a maximálně přeneseným 1 GB za měsíc. Pro zpracování HTTP požadavků slouží skripty vytvořené v jazyce PHP.

Výhodou jazyka PHP je nativní podpora pro připojení k MySQL databázi a také možnost snadné modifikovatelnosti. Jelikož se jedná o jazyk interpretovaný, je překlad prováděn vždy až při spuštění. Pro změnu chování tedy stačí prostřednictvím FTP připojení modifikovat daný soubor se skriptem a změna se projeví ihned při dalším spuštění. Podobně jako jazyk C++ nabízí jak procedurální, tak objektově orientovaný přístup k programování. Součástí HTTP požadavku mohou být předávána data 2 způsoby. Metoda POST předává data uvnitř těla paketu, zatímco GET přenáší data textově jako součást URL adresy. Díky snadné implementaci na straně mikrokontroléru byla zvolena GET metoda. V ní je umožněno předávat více metaproměnných, jež jsou od zbylé části URL adresy odděleny pomocí znaku "?" a mezi sebou znakem "&". V praxi tedy URL může vypadat takto:

`http://www.domena.cz/index.php?promenna1=hodnota1\&promenna2=hodnota2.`

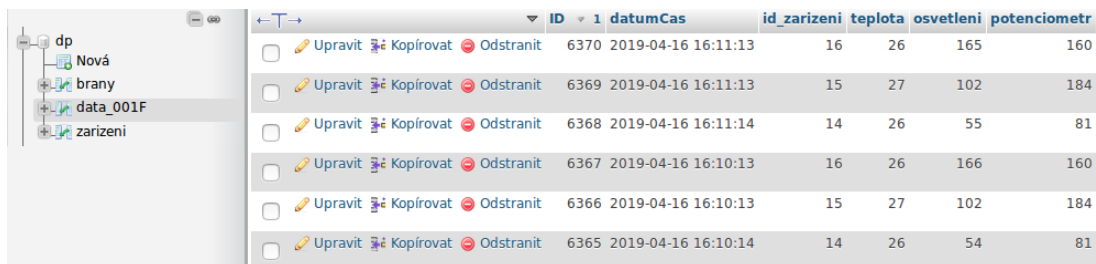


Data z komunikátoru jsou přenášena v rámci jedné metaproměnné *data* formou textového zobrazení hexadecimálních hodnot jednotlivých bajtů. Výsledná přenesená zpráva má dvojnásobnou velikost oproti původnímu paketu, avšak je to nezbytné z důvodu omezení při přenášení dat metodou GET z komunikátoru. Pro co nejuniverzálnější použití byla navržena struktura databáze, jež je zobrazená na obrázku 39. První tabulka *Zařízení* slouží jako přehled všech měřících uzlů jež přispívají svými daty do databáze. V každém záznamu je uvedeno výrobní číslo IQRF modulu MID, dále hardwarový profil HWPID a adresa v síti NADR. Tato tabulka zároveň slouží jako zdroj cizího klíče pro datové tabulky. Díky němu je každý datový záznam spřažený s měřícím zařízením odkud vzešel. Identifikátor hardwarového profilu slouží pro rozlišení druhu zařízení, s kterým úzce souvisí charakter a množství dat jež poskytuje. Databáze byla navržena modulárně tak, že pro každý typ zařízení podle HWPID, je vytvořena jedna tabulka, jejíž počet sloupců a datové typy korespondují s měřenými veličinami. Jak bude dále popsáno pro účely testování byl použit modul DDC-SE-01, ten měří teplotu, osvětlenost fotorezistoru a natočení potenciometru. Všechny 3 hodnoty jsou v 8bitovém rozlišení. Tabulka *data\_001F* tedy kromě časové značky a cizího klíče, který odkazuje do tabulky zařízení, obsahuje sloupce pro zmíněné veličiny. Jelikož je předpoklad použití většího množství navržených komunikátorů, je navržena třetí tabulka *Brány*, do které jsou ukládány informace při posledním připojení komunikátoru. Jedná se o stupeň nabití akumulátoru a časová značka posledního připojení.



Obrázek 39: Diagram navrhované databáze

Každý druh zařízení, rozlišený HWPID, musí mít v rámci PHP skriptu vytvořenou svou vlastní třídu. Zpracování příchozích zpráv bylo vytvořeno maximálně univerzálně. Každá příchozí měřená data jsou opatřena výrobním číslem modulu a právě zmíněným HWPID. Na základě něj je vybrána příslušná třída, která zajišťuje korektní interpretaci a uložení do databáze. Při vytváření instance dané třídy je jako vstupní argument konstruktoru předáno pole dat, uvnitř pak dojde k interpretaci jednotlivé měřené hodnoty. Díky dědění z abstraktní třídy *Device* je nutné implementovat metodu *DBinsert*, která zajistí zápis do správné tabulky databáze.



	ID	datumCas	id_zarizeni	teplota	osvetleni	potenciometr
<input type="checkbox"/> Upravit Kopírovat Odstranit	6370	2019-04-16 16:11:13	16	26	165	160
<input type="checkbox"/> Upravit Kopírovat Odstranit	6369	2019-04-16 16:11:13	15	27	102	184
<input type="checkbox"/> Upravit Kopírovat Odstranit	6368	2019-04-16 16:11:14	14	26	55	81
<input type="checkbox"/> Upravit Kopírovat Odstranit	6367	2019-04-16 16:10:13	16	26	166	160
<input type="checkbox"/> Upravit Kopírovat Odstranit	6366	2019-04-16 16:10:13	15	27	102	184
<input type="checkbox"/> Upravit Kopírovat Odstranit	6365	2019-04-16 16:10:14	14	26	54	81

Obrázek 40: Ukazáka výpisu tabulky MySQL databáze

## 8 Návrh servisního komunikačního protokolu

Tato kapitola se v první části zabývá popisem návrhu komunikačního protokolu, sloužící pro servisní zásahy a malé změny v chování komunikátoru bez nutnosti modifikace firmwaru. V druhé části je popsána vytvořená jednoduchá aplikace, která tento protokol implementuje.

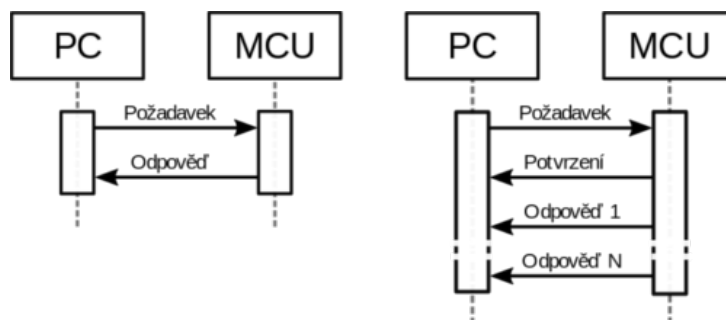
### 8.1 Návrh servisního protokolu

V předchozích kapitolách bylo zmíněno, že pro komunikaci komunikátoru s počítačem je využita sběrnice USB. Konkrétněji USB jako virtuální sériová linka. Po připojení se tedy komunikátor registruje v operačním systému jako COM port. V aktuálních verzích počítačových operačních systémech je nově nalezenému hardwaru automaticky přiřazen generický CDC ovladač. Z pohledu aplikace, jež má tento protokol implementovat, stačí pouze skrz API OS přistupovat k portu, který má odpovídající VID a PID.



Obrázek 41: Zobrazení struktury paketu

Navržený protokol je bajtově orientovaný, čili není čitelný z běžného sériového terminálu. Každý paket se skládá ze 3 hlavních částí. Jak je patrné z obrázku 41 obsahuje vždy příkaz, který identifikuje účel daného paketu, délku dat a samotná data. Z důvodu omezení, které vyplývá z ovladače USB na mikrokontroléru, může mít každý paket maximálně 64 bajtů.



Obrázek 42: Zobrazení dvou možných scénářů komunikace

Mikrokontrolér v roli hosta nikdy neodesílá data bez vyzvání. Tak jako je zobrazeno na obrázku 42, navržený protokol připouští 2 scénáře průběhu komunikace. První z nich představuje typickou komunikaci požadavek-odpověď (tzv. *request-response*). Avšak v případě, že požadavek je zaměřen na data, která mikrokontrolér momentálně nemá k dispozici, je použit druhý scénář požadavek-potvrzení-odpověď. Při němž mikrokontrolér hostiteli odesílá pouze potvrzení bez dat, načež provede kroky k jejich získání. Jakmile jsou data připravena, jsou odeslána jako asynchronní

odpověď. Jsou-li však tyto data rozsáhlá a nevlezu do jednoho paketu, jsou odeslána jako posloupnost asynchronních paketů. Hostitel dopředu zná, které zprávy vyžadují asynchronní odpovědi, u nich po doručení potvrzení přejde do čekání na odpověď.

Výpis kódu uveden níže ukazuje definici datových typů, jež jsou použity ve firmwaru mikrokontroléru pro obsluhu servisního protokolu. Pro příkaz i délku dat jsou vyhrazeny dva bajty. Přestože by stačil bajt jeden, je toto uspořádání nezbytné z toho důvodu, aby data začínaly na adrese zarovnané čtyřmi. Procesorové jádro ARM<sup>®</sup> Cortex<sup>®</sup> M0+ neumožňuje nezarovnané více bajtové přístupy do paměti. Při pokusu o tento přístup dojde v jádře k chybě a program skočí do tzv. *Hard faultu*.

Výpis kódu uveden níže ukazuje definici datových typů, jež jsou použity ve firmwaru mikrokontroléru pro obsluhu servisního protokolu. Pro příkaz i délku dat jsou vyhrazeny dva bajty. Přestože by stačil bajt jeden, je toto uspořádání nezbytné z toho důvodu, aby data začínaly na adrese zarovnané čtyřmi. Procesorové jádro ARM<sup>®</sup> Cortex<sup>®</sup> M0+ neumožňuje nezarovnané více bajtové přístupy do paměti. Při pokusu o tento přístup dojde v jádře k chybě a program skočí do tzv. *Hard faultu*.

---

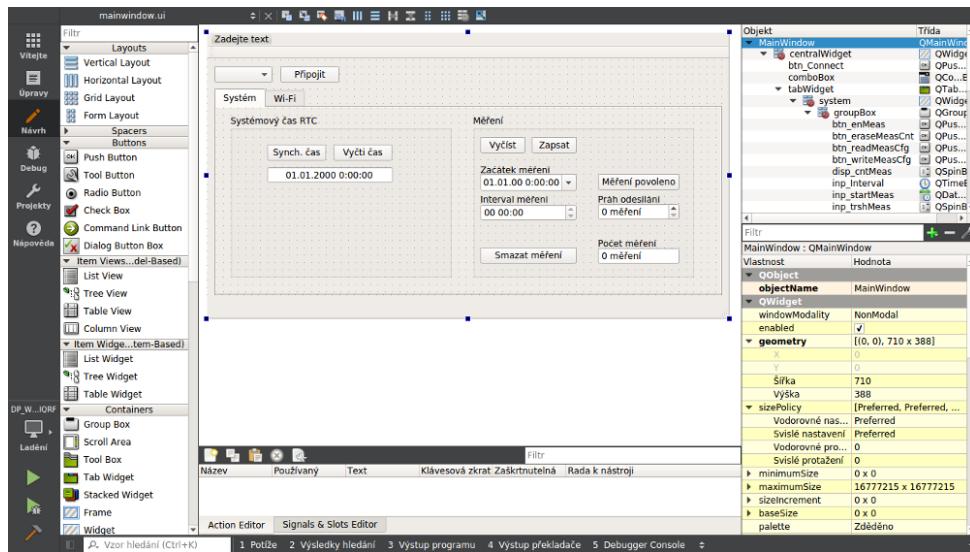
```
typedef enum
{
    USB_CMD_TestConn = 10,          // žádná data
    USB_CMD_ReadTimne = 11,         // 4 bajty, time_t z RTC
    USB_CMD_WriteTime = 12,         // 4 bajty, time_t pro RTC
    USB_CMD_Wifi_ReadInfo = 15,     // žádná data
    USB_CMD_Wifi_ReadInfo_Res = 21, // více paketů, struktura nastavení Wi-Fi
    USB_CMD_Wifi_ScanAP = 16,       // žádná data
    USB_CMD_Wifi_ScanAP_Res = 17,   // více paketů, údaje o AP
    USB_CMD_ReadMeasCfg = 18,       // 15 bajtů, čtení konfigurace
    USB_CMD_WriteMeasCfg = 19,      // 15 bajtů, zápis konfigurace
    USB_CMD_EraseMeas = 20,         // žádná data
    USB_CMD_Error = 65535           // žádná data
}USB_CMD_t;
typedef struct
{
    USB_CMD_t cmd;
    uint16_t len;
    uint8_t data[60];
}
```

---

Výpis 9: Datové typy použité pro komunikační protokol

## 8.2 Realizace servisní aplikace

Navržený protokol byl implementován do jednoduché aplikace. Ta byla naprogramována v jazyce C++ pod Qt frameworkem. Velkou výhodou tohoto řešení je možnost vytvářet multiplatformní aplikace, bez vytváření více vývojových větví. Qt framework poskytuje rozsáhlé knihovny pro všechny běžně používané operační systémy. Oprostití-li se programátor od přímého používání API operačního systému, ale využije právě zmíněné knihovny, je kód plně přenositelný a bez úprav zkompileovatelný na kterékoliv jiné platformě (Linux, macOS, Windows). Spolu s těmito knihovnami vývojáři poskytují rovněž vývojové prostředí Qt Creator.



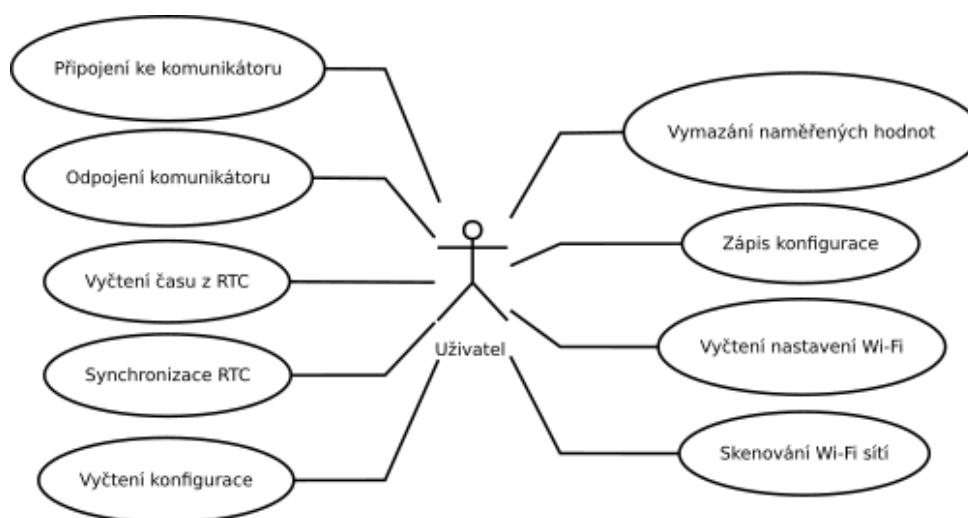
Obrázek 43: Ukázka vývojového prostředí Qt Creator

Vývojové studio nabízí, mimo jiné, programátorům návrhář formulářových aplikací, kde lze pomocí WYSIWYG editoru navolit vzhled vytvářené formulářové aplikace. Vybrat si je možné z široké nabídky předpřipravených prvků jako jsou tlačítka, textová pole, tabulky, či záložky. Formulář vytvořené servisní aplikace se skládá z rozbalovacího menu pro výběr sériového portu, tlačítka pro připojení/odpojení a dvou záložek. První záložka nabízí možnost načtení a synchronizace času v RTC komunikátoru spolu s možností vyčtení a změny konfigurace. V druhé záložce je možné spustit skenování dostupných Wi-Fi sítí a vyčtení aktuálního nastavení Wi-Fi modulu.

Jazyk C++ vznikl jako rozšíření jazyka C. Mimo jiné přibyla podpora objektově orientovaného programování, avšak na rozdíl od C# lze tvořit program i procedurálním programovacím stylem. Vytvořená aplikace se skládá ze dvou tříd. První třída MainWindow slouží k zobrazení okna formuláře, pro použití grafických prvků z knihoven Qt dědí od QMainWindow. Druhá použitá třída byla vytvořena pro zprostředkování komunikace navrženým protokolem. Její součástí je instance třídy pro ovládání sériové linky. Ve vnějším rozhraní třídy jsou metody pro vyčtení

seznamu sériových portů, jejichž identifikátory odpovídají komunikátoru, dále metody pro připojení, odpojení, odeslání zprávy a kontrolu, zda je zařízení připojeno. Tato třída si v případě delší neaktivity hlídá připojení odesláním dotazů na test spojení. Dojde-li k odpojení komunikátoru třída USB to detekuje a vyvolá příslušnou událost. Jazyk C++ nemá nativní mechanismy pro předávání událostí nadřazeným třídám, tak jako je tomu v případě jazyka C#. Ovšem při použití Qt frameworku, konkrétněji knihovny *QObject* lze využít systému signálů a slotů. Třída USB tedy má 2 veřejné signály (doručená zpráva a jiná událost). V místě použití se pomocí funkce *connect* zajistí spojení vybraného signálu použité třídy s místní metodou - slotem.

Ve třídě *MainWindow* je tedy vytvořena instance třídy *USB*, skrze kterou probíhá veškerá komunikace s komunikátorem. Jednotlivá tlačítka poté ve svých callback funkcích vyvolávají odeslání příslušné zprávy. Jakmile dojde od mikrokontroléru odpověď, přes zmíněný signál, je zavolána metoda třídy *MainWindow*. Podle příkazového identifikátoru je následně s došlými daty naloženo a jsou vizualizována.

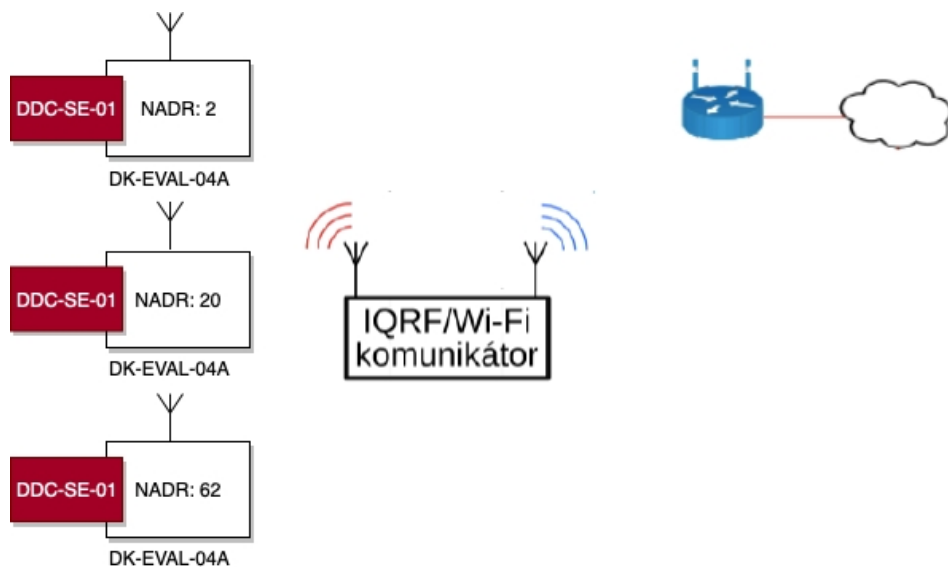


Obrázek 44: UML diagram případu užití pro vytvořenou aplikaci

## 9 Testování komunikátoru

Pro účely testování zařízení s ohledem na spotřebu elektrické energie byla vytvořena malá IQRF síť. Jako měřicí uzly byly využity 3 moduly DK-EVAL-04A. Samotný IQRF modul DCTR-72D, osazený ve vývojové základně, umožňuje měřit teplotu, avšak pro účely testování byl k základně připojen senzorický vývojový modul DDC-SE-01A. Ten obsahuje hned dvojici teplotních senzorů mezi nimiž je možné přepínat, nemohou však fungovat simultánně. Prvním z nich je Dallas DS18B20, jež komunikuje s mikrokontrolérem uvnitř IQRF prostřednictvím sběrnice 1-Wire. Druhým připojeným teplotním senzorem skrze I<sup>2</sup>C je MCP9802A3. Jako další zdroje měřitelných dat na senzorickém modulu slouží fotorezistor a potenciometr, zapojené jako napěťový dělič. Získaná napětí obou zdrojů jsou přivedena na operační zesilovače, jež slouží jako napěťové sledovače. To zabraňuje ovlivnění dělicích poměrů vstupní impedancí AČ převodníku, kam jsou oba zmíněné napěťové signály přivedeny.

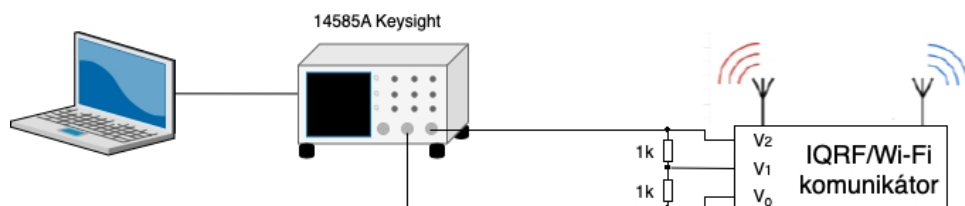
Před vytvořením IQMESH sítě byl do jednotlivých měřicích uzlů nahrán Custom DPA handler, jež slouží jako příklad pro použití DCC-SE-01A. Ten při zaslání DPA požadavku na periferii PNUM 0x20 a příkazem PCMD 0x31 zprostředkuje odpověď s třemi datovými bajty. První z nich odpovídá teplotě ve stupních Celsia, hodnota druhého bajtu je nepřímo úměrná míře osvětlenosti fotorezistoru. Třetí bajt reprezentuje otočení hmatníku na potenciometru. V celé IQMESH síti byly použity hardwarové profily pro LP mód IQRF modulů.



Obrázek 45: Znázornění vytvořené testovací sítě

Navržený komunikátor v této síti vystupuje jako koordinátor a posílá jednotlivým měřicím uzlům požadavky na data. Napájení komunikátoru je místo akumulátoru realizováno z přístroje

Keysight 14585A. Ten slouží jako nastavitelný napájecí zdroj s velmi přesným měřením spotřeby. Aby bylo simulován skutečný akumulátor, byl vybrán režim napětového zdroje s hodnotou, jež odpovídá jmenovitému napětí akumulátoru 7,4 V. Vnitřní impedance zdroje byla ponechána na výchozí hodnotě  $0\ \Omega$ . Bateriový monitor však očekává připojení dvou sériově řazených článků. To mělo za následek vytvoření napětového děliče s použitím dvou  $1\text{k}\Omega$  rezistorů pro získání středového napětí. Dělič však zvyšuje odběr elektrického proudu, který přes něj příčně teče. Tuto skutečnost bylo nutné ve výsledcích zohlednit.



Obrázek 46: Schéma zapojení použité při měření spotřeby

Měření spotřeby bylo realizováno pro běžný funkční režim a také pro režim se sníženou spotřebou. V obou uvedených případech se sbírala data ze sítě, jež je popsána výše, v minutovém intervalu. U režimu nízké spotřeby byl navíc ještě práh pro odesílání dat nastaven na hodnotu 14, čili každých 5 minut došlo k odeslání naměřených dat.

Tabulka 3: Tabulka změřených hodnot spotřeby

Režim	Změřená průměrná spotřeba	Odběr děliče	Průměrná spotřeba komunikátoru	Odhadovaná doba provozu
Běžný	77 mA	3,7 mA	72,3 mA	3d 16,52h
Úsporný	15,9 mA	3,7 mA	12,2 mA	21d 20,59h

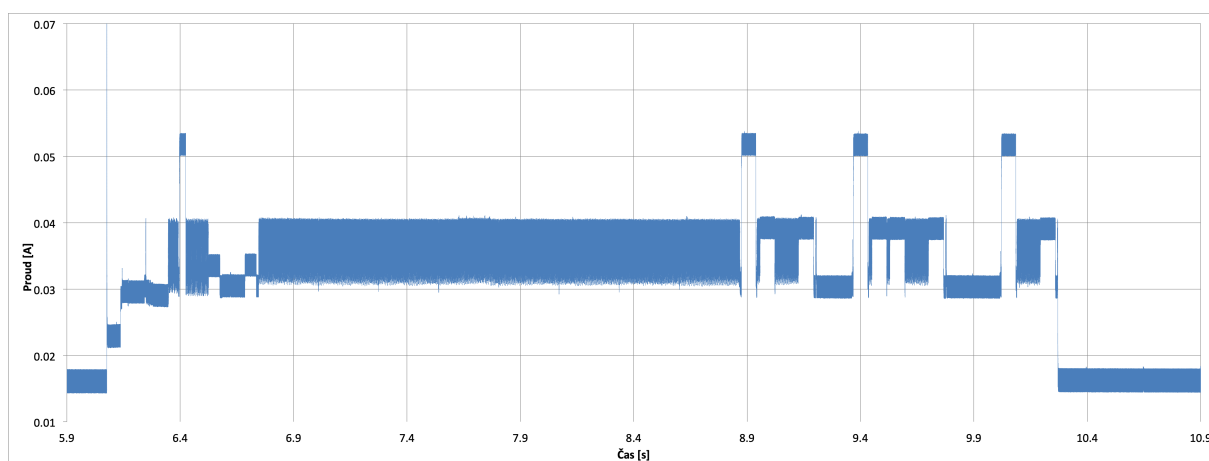
Vydeme-li ze jmenovité kapacity akumulátoru 6400 mAh, tak při zanedbání samovybíjení je podle výsledků měření teoretická doba běhu na akumulátor 3 dny a 16,52 hodin v běžném režimu. Při přepnutí do úsporného běhu se teoretická doba provozu prodlouží až na 21 dní a 20,59 hodin. Spotřeba v úsporném režimu ovšem silně závisí na IQMESH síti, ze které získává data, na počtu uzlů, jejich topologii a v neposlední řadě také na četnosti vyčítání dat. Obecně platí, že čím méně času měření zabere, tím méně bylo spotřebováno elektrické energie.

Na obrázku 47 je znázorněn časový průběh odběru proudu při sběru dat z IQMESH sítě. Je z něj dobře patrný okamžik zapnutí napájení nejprve pro FRAM a následně pro IQRf, tak jak je to popsáno v kapitole 6. Po naběhnutí následuje vyčítání dat ze všech 3 měřících uzlů jež má komunikátor uložen v mapě.

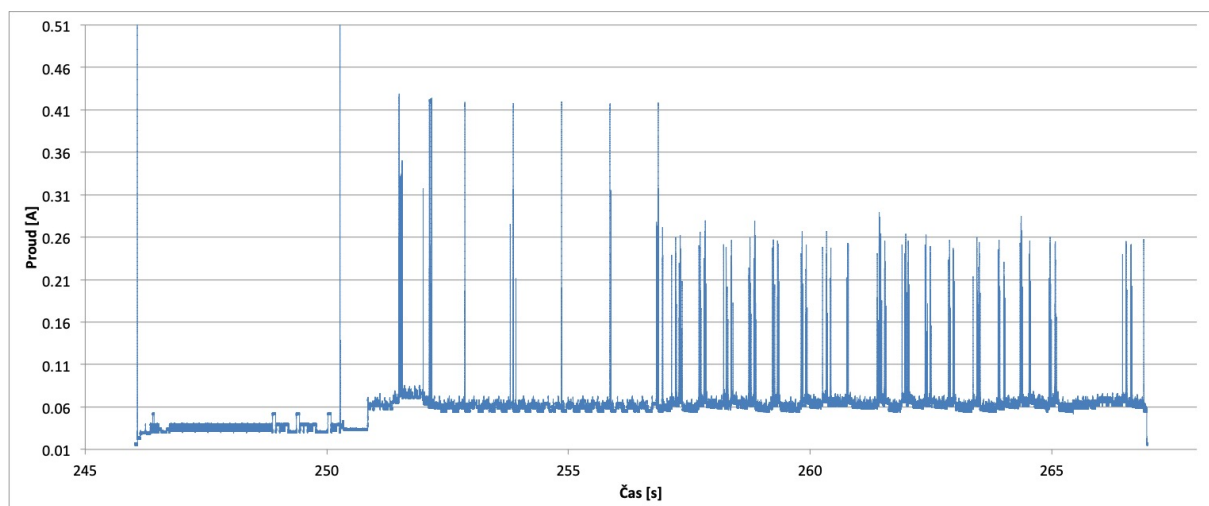
Obrázek 48 zobrazuje mimo sběr dat také odeslání hodnot ze zásobníku na server v úsporném režimu. Dobře patrné jsou okamžiky, kdy dochází k zapnutí napájení pro IQRf, respektive Wi-



Fi modul. Charakterizují je vysoké špičkové odběry způsobené nabíjením kondenzátorů. První část průběhu, přibližně do času 251 s, je tvarově totožná s předchozím obrázkem. Následně je vyhodnoceno, že počet měření v zásobníku překročil nastavený práh a dojde k zapnutí Wi-Fi modulu. Po vnitřní inicializaci, je provedeno připojení k Wi-Fi síti a k zahájení odesílání dat. Z výše zmíněného grafu je rovněž zřejmé, že došlo k odeslání 15 měření, jež byly na konci doplněny stavovými informacemi.



Obrázek 47: Časový průběh spotřeby při sběru dat z IQRF v úsporném režimu



Obrázek 48: Časový průběh spotřeby při sběru a odesílání dat v úsporném režimu

## 10 Závěr

Hlavní náplní této diplomové práce byl návrh s následnou realizací IQRf-Wi-Fi komunikátoru s ohledem na spotřebu elektrické energie. Samotnému návrhu a realizaci předcházely teoretický rozbor použitých technologií (IQRf a Wi-Fi), rodiny mikrokontrolérů NXP Kinetis® L a možnostmi přenosu dat z IQMESH sítě na datové úložiště.

V rámci této práce byl navržen elektronické zapojení komunikátoru. Během návrhu byl brán zřetel na spotřebu elektrické energie z důvodu možného umístění v prostorech bez možnosti stálého napájení. Po úspěšné realizaci návrhu v podobě DPS byl vytvořen firmware. Ten se sestává z několika režimů svého fungování. Nejprve byly implementovány servisní tunelové režimy pro přímý přístup k IQRf, potažmo k Wi-Fi modulu skrze micro USB konektor typu B. Úspěšně byl rovněž vytvořen běžný režim pro sběr dat a konfiguraci zařízení. Čtvrtým funkčním módem je docíleno sběru a odesílání dat za použití sníženého množství elektrické energie. Do firmwaru byl implementován zásobník na měřená data. Ten je využit jak v případě momentální nedostupnosti Wi-Fi sítě, tak i při úsporném režimu, kdy se data na úložiště odesílají ve větších intervalech než v jakých jsou sbírána. Další funkcionalitu představuje možnost automatického mapování IQRf sítě a následné uložení mapy do energeticky nezávislé paměti. Pro možnost změny konfigurace byl navržen a následně v podobě počítačové aplikace implementován servisní komunikační protokol. Tento softwarový nástroj byl vytvořen v jazyce C++ s využitím Qt frameworku zajišťující přenositelnost zdrojových kódů napříč běžně používanými operačními systémy. V dalším kroku bylo realizováno datové úložiště kombinací webového serveru a MySQL databáze. Během návrhu byl brán zřetel na zajištění funkčnosti s jakýmkoliv měřicím zařízením, jenž slouží jak zdroj dat.

Testování zhotoveného zařízení spočívalo ve vytvoření IQRf sítě s třemi měřicími uzly, přičemž každý uzel poskytoval trojici hodnot, spuštění sběru dat a měření spotřeby pomocí přístroje Keysight 14585A. Interval sběru dat byl pro snadnou měřitelnost spotřeby stanoven na jednu minutu. V případě nízkoodběrového režimu bylo odesílání dat prováděno co 5 minut. Běžný funkční režim po 10 minutách vykazoval průměrnou spotřebu 72,3 mA, což při plném nabití akumulátoru odpovídá v ideální případě výdrži 3 dnů a 16,52 hodin. Energeticky výhodný režim při stejné době měření vykazoval průměrný odběr 12,2 mA, odpovídající době provozu 21 dnů a 20,59 hodin. Tyto hodnoty jsou ovšem silně závislé jednak na četnosti sběru dat, ale také na počtu měřicích uzlů. Pro porovnávání, z katalogového listu IQRf-Wi-Fi brány od firmy Microrisc, s.r.o., lze vyčíst udávanou spotřebu při běhu 250 mA a v režimu spánku 85 mA. Energetická úspora v porovnání s komerčně dostupným řešením je tedy značná.

## Literatura

- [1] Energeticky soběstačný bezdrátový modul pro senzorické aplikace  
*JACKO, Róbert* . Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. Dostupné z: <http://hdl.net/11012/32913>.
- [2] Algoritmy číslicového zpracování signálů.  
*BRTNÍK, Bohumil a David MATOUŠEK*.. Praha: BEN - technická literatura, 2011. ISBN 978-80-7300-400-2.
- [3] . Velký průvodce protokoly TCP/IP a systémem DNS  
*DOSTÁLEK, Libor a Alena KABELOVÁ* . 2. vyd. Computer Press Praha, 2000. 426 s. ISBN 80-7226-323-4.
- [4] KL28Z Reference Manual  
*NXP Semiconductors*.. 2016 [online] Dostupné z: <http://www.nxp.com>.
- [5] IQRf Quick Start Guide For IQRf OS v3.07D and higher  
*MICRORISC s.r.o.*. 2017 [online] Dostupné z: <http://iqrif.org>.
- [6] IQRf OS Operating System Version 4.00D for ((DC)TR-7xD - User's Guide  
*MICRORISC s.r.o.*. 2017 [online] Dostupné z: <http://iqrif.org>.
- [7] IQRf DPA Framework Technical Guide v3.00  
*MICRORISC s.r.o.*. 2017. [online] Dostupné z: <http://www.iqrif.org>.
- [8] IQRf Cloud Technical guide  
*MICRORISC s.r.o.*. 2015 [online] Dostupné z: <http://iqrif.org>.
- [9] Limitations of Ad Hoc Mode Wireless Networking.  
*Lifewire* [online]. 2016[cit. 03.01.2018]. Dostupné z: <https://www.lifewire.com/limitations-of-ad-hoc-mode-networking-818294>.
- [10] RN-131G & RN-131C 802.11 b/g Wireless LAN Module.  
*Microchip* [online]. [cit. 2018-04-23]. Dostupné z: <https://www.microchip.com/wwwproducts/en/RN131#datasheet-toggle>
- [11] Vývoj a implementace SW pro DCS v4 (2017), Dílčí výzkumná brána k projektu MPO FV10706 Výzkum a vývoj inteligentní sítě seznorů pro monitorování bezpečnosti záchytných sítí a bariér pro potřeby geotechniky. *STANKUŠ, Martin*.
- [12] Všeobecné oprávnění č. VO-R/12/09.2010-12 k využívání rádiových kmitočtů a k provozování zařízení pro širokopásmový přenos dat v pásmech 2,4 GHz až 66 GHz.  
*Český telekomunikační úřad* [online]. [cit. 2018-04-23] Dostupné z: <https://www.ctu.cz/>

- [13] DPA  
*MICRORISC s.r.o.* [online]. [cit. 2018-04-23] Dostupné z:  
<https://www.iqrf.org/technology/dpa>
- [14] Kinetis® L Series: Ultra-Low Power Microcontrollers (MCUs) based on Arm® Cortex®-M0+ Core  
*NXP Semiconductors.* [online]. [cit. 2019-01-30] Dostupné z: <https://www.nxp.com>
- [15] TR-76D series  
*MICRORISC s.r.o.* [online]. [cit. 2019-01-30] Dostupné z  
<https://www.iqrf.org/products/transceivers/tr-76d>
- [16] BQ28Z610 Technical reference  
*Texas Instruments* [online]. [cit 2018-12-20] Dostupné z <http://www.ti.com/>
- [17] CY15B104Q Datasheet  
*Cypress* [online]. [cit. 2018-12-20] Dostupné z <https://www.cypress.com/>
- [18] WiFly Command Reference Manual  
*Microchip* [online]. [cit. 2018-12-20] Dostupné z <https://www.microchip.com>

## A Schéma zapojení a deska plošných spojů

Projekt pro software Eagle je ve formě elektronické přílohy uložen na CD ve složce */eagle*.

## B Firmware

Projekt pro Atollic TrueSTUDIO se všemi zdrojovými kódy je ve formě elektronické přílohy uložen na CD ve složce */firmware*.

## C Servisní aplikace

Projekt pro Qt Creator se všemi zdrojovými kódy je ve formě elektronické přílohy uložen na CD ve složce */aplikace*.

## D PHP skripty a databáze

Veškeré vytvořené PHP skripty spolu se zálohou databáze jsou ve formě elektronické přílohy uloženy na CD ve složce */server*.